PHYSICS-GUIDED LEARNING AND SURROGATE MODELING FOR STRUCTURAL DESIGN AND HEALTH MONITORING APPLICATIONS

By

Ali Irmak Özdağlı

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

December 17, 2022

Nashville, Tennessee

Approved:

Xenofon Koutsoukos, Ph.D.

Janos Sztipanovits, Ph.D.

Gautam Biswas, Ph.D.

Akos Ledeczi, Ph.D.

Sankaran Mahadevan, Ph.D.

ACKNOWLEDGMENTS

My most sincere gratitude and appreciation goes to my advisor, Dr. Xenofon Koutsoukos for his continuous guidance, support and patience. Without his advice in all stages of my studies, this dissertation would not have been a reality. Dr. Koutsoukos has my deepest gratitude without any reservation! I also want to thank my committee members, Dr. Janos Sztipanovits, Dr. Gautam Biswas, Dr. Akos Ledeczi and Dr. Sankaran Mahadevan for their valuable input to this dissertation.

I also want to acknowledge my fellow graduate students, Feiyang, Chandreyee, and Nick. Their company made my life always full of happiness and fun.

Finally, I would like to thank my family for their love and support during my journey. I would also like to thank my beloved Judyvin. Without her support, I would not be able to materialize this dissertation.

The financial support of this research is provided in part by the U.S. National Science Foundation under Grant CNS-1238959, the National Institute of Standards and Technology under Grant 70NANB18H198, the Defense Advanced Research Projects Agency through contract number FA8750-18-C-0089 and FA8750-20-C-0537, and IBM Graduate Fellowship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Motivation

In the last few decades, structural health monitoring (SHM) has gained a lot of momentum as a means of detecting and localizing damages Sohn et al. (2002). The introduction of machine learning (ML) into SHM enabled further refinement as mature pattern recognition techniques provide higher accuracy in recognizing structural damages compared to traditional methods (Farrar and Worden, 2012). Among many ML applications, supervised methods are particularly useful (Kiranyaz et al., 2019). Especially, when coupled with artificial neural networks, supervised learning offers promising results for damage detection and localization (Park et al., 2009; Dackermann et al., 2013; Nick et al., 2015).

A majority of supervised SHM applications assume that the data used for training the damage condition classifier has the same distribution as the testing data. However, this assumption is problematic. It is unrealistic that one can obtain data belonging to a particular damage condition without actually harming the integrity of the structure before its service (Lu et al., 2016; Gardner et al., 2020). We can generate a labeled data set using a representative physics-based finite-element model (FEM) where introducing damages is a more cost-effective approach and we can train a black-box machine learning model using this data. On the other hand, calibrating a large set of FEM parameters for complex systems to achieve accurate physical behavior is often computationally exhaustive and at times infeasible Zhang et al. (2020). As a result of this, the analytical representation inherits some modeling error. In this case, it is expected that the ML algorithm will fail to perform effectively during testing since the simulation training data and experimental testing data are statistically divergent (Gardner et al., 2020). To address this drawback of data-driven black-box algorithms, the inference should incorporate either a knowledge transfer from simulation data to experimental data or domain-specific physical knowledge, or both.

In engineering, the performance of a design is usually validated through a comprehensive analysis. As mentioned above, this analysis may involve complex computer simulations such as FEMs as they are known to predict the behavior of physical systems to some fidelity and to provide valuable insight into the design process. On the other hand, running high-fidelity simulations is computationally demanding. Especially, in an iterative design process, where broad design space is explored, running simulations repeatedly is exhaustive and extremely time consuming (Audet et al., 2000). Instead of this cost-prohibitive approach, it is necessary to develop a surrogate model that is capable of approximating the simulation outputs. Naturally, we expect

this surrogate model to evaluate given inputs relatively faster compared to a high-fidelity simulation making the design process accelerated and more affordable where computational budget is limited.

1.2 Research Challenges

As discussed above, for typical damage identification and classification tasks, our access to labeled experimental data is limited, yet we can create abundant simulation data. While the domain adaptation (DA) approach is capable of transferring knowledge from the simulation domain to the experiment domain, this process may suffer from negative transfer learning as well. In this context, negative learning refers to the decline in classification performance due to poor knowledge transfer. As a result of this, we may observe a performance degradation in predictions for some classes.

Another alternative for reducing the statistical differences between simulation and experiment domains is infusing the machine learning algorithm with physics-based knowledge that can generalize well over both domains and guiding the learning process. One important challenge for physics-guided learning (PGL) is selecting the physical knowledge that is most significant for the task and the method to introduce this knowledge into the training process. This work explores the use of modal parameters as intermediate layers for damage localization tasks. However, our findings show that PGL architecture may suffer from a peculiar issue called the choking effect resulting in a classification quality degradation when the number of intermediate layer parameters is small. Accordingly, our objective should be to improve the prediction performance by extending PGL's learning capability and capacity. To achieve this aim, we should consider integrating DA into PGL.

Another challenge attributed to machine learning, in particular, neural networks, is the lack of interpretability. PGL uses human-recognizable physics-based knowledge during the training process. As a natural result of this property, PGL has the capability to relate the damage condition of the target structure to the physical parameters. However, it is still a challenge how to establish a relationship between predicted damage condition and the physics of the system that will reveal the most informative interpretation.

Another component of the work presented here is concerned with generating designs through surrogate modeling. A majority of neural network based surrogate modeling methods usually applies a data-driven black-box modeling. While the black-box techniques are very efficient and accurate as surrogates in predicting the responses, they may often fail in generalizing over the less explored design areas. Moreover, they may not be transparent enough to provide an explanation for the user exploring the design space. This research seeks to address these challenges by integrating PGL into the surrogate model and improve the model explainability.

1.3 Contributions

In this proposal, our core work mainly focuses on the following problems: (i) domain adaptation problems where target domain data is very limited, (ii) the integration of physics-based knowledge into the machine learning process, (iii) interpretability of deep networks through physics-guided learning, and (iv) the development of surrogate models using physics-guided learning to eliminate the high cost of complex computational simulations.

The main contributions are listed below:

Chapter 3

In this chapter, we proposed an approach for detecting novelties in structures using their model properties such as mode shapes and natural frequencies. The main contributions are:

- We proposed an end-to-end architecture to detect damage under environmental uncertainty using machine learning. The proposed approach streamlines (a) collection of structural response data, (b) modal analysis using system identification, (c) auto-encoder, and (d) novelty detection. The proposed system aims to extract latent features of accessible modal parameters such as natural frequencies and mode shapes measured at undamaged target structure under temperature uncertainty and to reconstruct a new representation of these features that is similar to the original using auto-encoders. The deviation between measured and reconstructed parameters, also known as the novelty index, is the essential information for detecting critical changes in the system.

- We evaluated our approach through simulations and experimental tests. The results demonstrate that the effectiveness of the damage detection under temperature variability improves significantly compared to the previous damage detection algorithms in the presence of environmental variability. Especially for small damages, the proposed algorithm performs better in discriminating system changes.

Chapter 4

In this chapter, we developed a domain adaptation approach to localize and classify damages in structures where the classifier has access to the labeled training (source) and unlabeled test (target) domain data, and the source and target domains are statistically different. The main contributions are:

- We proposed a domain adaptation method to form a domain-agnostic feature space that is capable of representing both source and target domains by implementing a domain-adversarial neural network. This neural network uses H-divergence criteria to minimize the discrepancy between the source and

target domain in a latent feature space. Compared to the popular methods such as Transfer Component Analysis (TCA), the domain adversarial network provides a better generalization across both domains.

- We evaluated the performance of the proposed method using (1) gearbox system data, and (2) a three-story experimental structure. We demonstrated the effectiveness of the domain adaptation by computing the damage classification accuracy for the unlabeled target data with and without domain adaptation. Furthermore, the performance gain of the domain adaptation over TCA is also shown. Overall, the results demonstrate that the domain adaption is a valid approach for SHM applications where access to labeled experimental data is limited.

Chapter 5

In this chapter, we propose PGL by integrating physics knowledge into machine learning and constraining the training process. In summary, the major contributions of this work are:

- We propose a PGL approach that combines the power of data-driven machine learning with physics-based models. Specifically, the proposed architecture integrates physical parameters extracted from the physics-based simulation into the neural network in the form of intermediate layers to constrain the learning process. To accommodate the intermediate layers, the architecture introduces physics-based loss into empirical loss function. PGL is especially useful for cases where it is challenging to develop a reliable black-box learning model due to the lack of training data originating from the real system. We can generate extensive training data using simulators. While this data may be statistically somewhat different than the true system behavior, PGL is encoded with domain knowledge, and it is capable of predicting the true system response using this encoding.

- We demonstrate the effectiveness of the proposed approach in two different use cases where the task is localizing and classifying the damage for a given set of structural responses. Here, we assume that we don't have access to the experimental structural responses or the damage classes. But we can generate responses and corresponding damage location labels using a numerical representation of the actual system. Our architecture is trained using this numerical data. Additionally, the architecture is infused with domain-specific physics knowledge by extracting modal properties such as natural frequencies and mode shapes from the numerical model and using these features in intermediate variable layers. Our results show that the accuracy for localizing the damage correctly improves significantly over black-box models.

- The physics-based intermediate layers improve learning during training. Another benefit of the intermediate layer is that it provides additional feature information during testing that was previously

not available for black-box models. In this way, our proposed architecture improves the explainability of the results since the intermediate layers expose valuable information that is highly relevant to the physics of the target structure. To demonstrate the effectiveness of this capability, we propose a method for analysis and interpretation of the intermediate results and their relevance to predicted classes. We show that our interpretability method can lay out the relationship between predicted modal properties within the intermediate layers and the predicted damage class for testing data effectively. Our findings indicate that the misclassified instances could be explained through the characterization of predicted natural frequencies.

Chapter 6

In this chapter, we extend the capability of PGL and DANN by fusing the two architectures together to improve generalization capability of deep networks. The main contributions are:

- We develop a multi-task deep learning architecture that ingests domain adaptation and physics-based domain specific knowledge into the training procedure.
- We evaluate the performance of this proposed method by comparing the damage classification accuracy to the black-box model, PGL architecture, DANN implementation, and the proposed PGL-DANN method.

Chapter 7

In this chapter, we propose a method for improving interpretability and explainability of deep learning architectures through physics-guided learning using layer-wise relevance propagation (LRP). In summary, the major contributions of this work are:

- We propose a set of relevance rules for LRP that is most significant (or human-recognizable) for explaining the relationship between modal properties (intermediate layers) and damage.
- We evaluate LRP results by qualitatively analyzing the relevances and comparing our findings to traditional qualitative approaches.

Chapter 8

In this chapter, we propose a PGL-driven surrogate modeling approach to generalize the surrogate over the design space The main contributions are:

- We propose a simple method to explore the design space for a domain specific application and to train a physics-guided surrogate model.

- We evaluate the surrogate model of various design using explainable AI through LRP.

1.4 Organization

The content of the proposal is organized as follows:

- Chapter 2 reviews the literature on novelty detection, domain adaptation, physics-guided learning, and surrogate models.
- Chapter 3 discusses the problem of detecting damage in structures using auto-encoders.
- Chapter 4 discusses the problem of domain adaptation towards damage localization when limited target data is available for training.
- Chapter 5 discusses the problem of physics-guided learning using intermediate value layers and the integration of domain adaptation into the learning phase.
- Chapter 6 discusses the problem of integrating physics-guided learning with domain adaptation.
- Chapter 7 discusses the problem of the interpretability for neural network models through physics-guided learning.
- Chapter 8 discusses the problem of physics-driven surrogate modeling.
- Chapter 9 concludes the dissertation and discusses the results in general.

CHAPTER 2

Related Work

Within the last decade, the research on fault detection and isolation (FDI) has gained momentum with the recent advancements in machine learning (ML). One of the grand challenges for FDI applications is access to complete training dataset covering a wide range of conditions of the target system (Widodo and Yang, 2007; Farrar and Worden, 2012; Stetco et al., 2019; Zhang and Sun, 2021). This problem is a major roadblock in developing efficient data-driven algorithms for diagnosing dynamic systems properly (Sadoughi and Hu, 2019). It is imperative to develop effective methodologies to mitigate the shortcomings of ML-based FDI. In this chapter, we review the related work in the area of physics-guided learning, domain adaption, and surrogate modeling. This chapter opens up with novelty detection which is the necessary first step for detecting faults in dynamic systems, discussed in Section 2.1. Section 2.3 reviews the work on the integration of physical knowledge into the data-driven ML to improve the overall performance and accuracy. Section 2.2 discusses the domain adaptation problems to treat the differences between source and target domains. Finally, Section 2.4 discusses the literature on surrogate modeling. Figure 2.1 summarizes the taxonomy of the work discussed in this chapter.

2.1 Novelty Detection

Within the context of FDI, novelty detection refers to the ML applications to automatically recognize the health state and detect faults on dynamic systems such as machines or structures. In this section, we first introduce the general process of novelty detection (see Figure 2.2) which involves prior steps such as data collection, and feature extraction. Then, we discuss the traditional novelty detection methods which do not utilize neural networks. finally, we review the work on neural network-based approaches. Conformal detection techniques are omitted since it is out of scope.

2.1.1 Data Collection and Feature Extraction

The first step for novelty detection is collecting data from the target structure. Typically, a variety of sensors are mounted on the target structure to gather measurements from the system continuously. Depending on the application, the type of the sensor is customized. The mainstream practice towards fault detection utilizes vibration data through accelerometers. For example, the health condition of gearbox systems (Phm Society, 2009) and civil structures (Caicedo et al., 2004) can be monitored using vibration data. Other novelty detection methods focusing on crack detection use acoustic emission data captured with piezo-like sensors. The

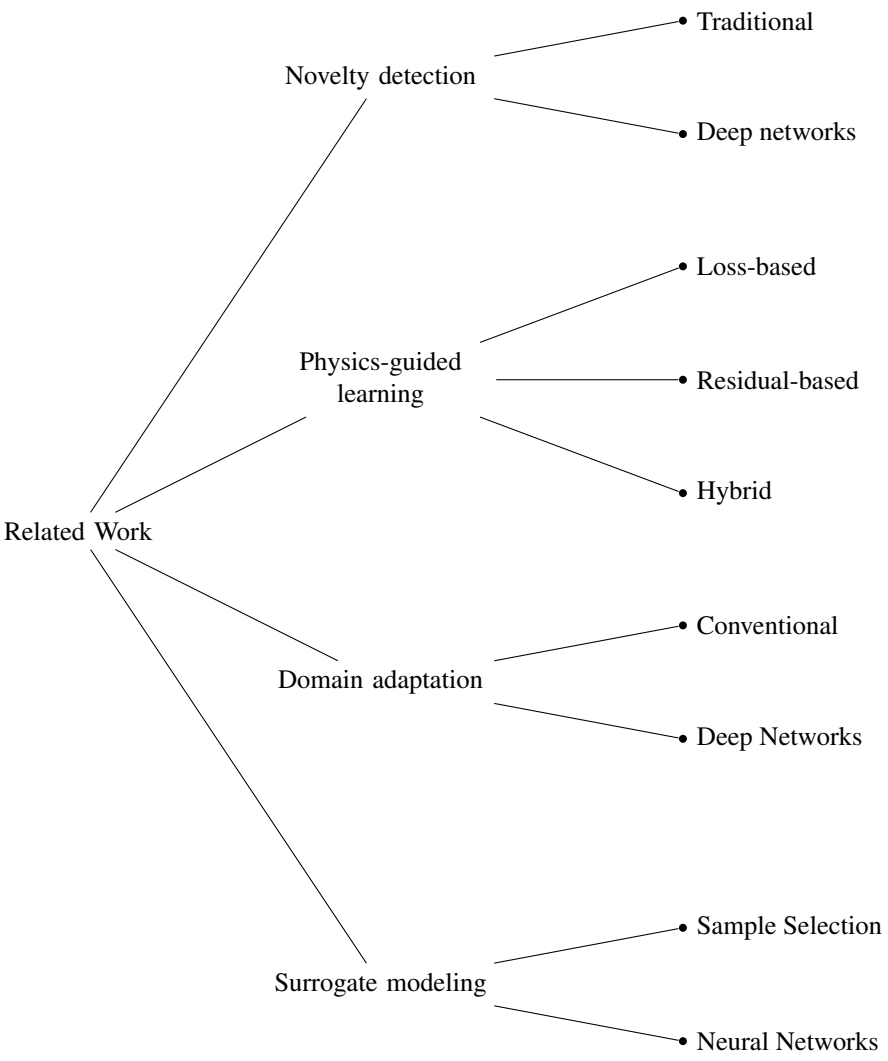


Figure 2.1: Summary of related work

Figure 2.2: Novelty detection

combustion faults in engines due to failure (Fuentes et al., 2020), dealignment in composite materials (Marec et al., 2008), and internal cracks in concrete (Mason et al., 2016) are some examples that utilize acoustic emission data.

The next important step is the feature extraction and selection process. The commonly used features are sensor data in time, frequency, or time-frequency domain. Time-domain features can be composed of unprocessed data, or statistical features such as mean, standard deviation, peak value, etc. (Lei et al., 2010). Frequency domain features are obtained through frequency domain analysis techniques such as operation modal analysis (Ozdagli and Koutsoukos, 2019). Lastly, time-frequency domain features are obtained through wavelet transformations (Chen and Zhan, 2008).

2.1.2 Traditional methods

Before the introduction of neural networks, the traditional ML-based approaches were utilized frequently to detect the damage in dynamic systems. These approaches can be categorized under two main classes: i) SVM; ii) other methods encapsulating k-Nearest Neighbor (kNN); decision trees (DT); Bayesian classifier (BC).

SVM (Cortes and Vapnik, 1995) is a supervised learning method mainly used to separate classes for a given dataset. SVM is very suitable for novelty detection since the problem can be converted into a healthy vs damaged classification problem according to the definition of SVM. There is a wide range of SVM applications that also integrate various optimization algorithms for better detection. Lei et al. (2020) summarize some of the SVM-based novelty detection.

kNN (Cover and Hart, 1967) is another supervised learning method for classification. kNN transforms the given data into a feature vector based on the Euclidean distance. Data that share similar features are expected to be close to each other such that they can be categorized under the same label. Lei and Zuo (2009) and Dong et al. (2017) studied the effectiveness of kNN approach in detecting damage for gear systems and bearings. Lei et al. (2020) argue that kNN algorithms are not popular for damage detection as they require heavy computational power for large datasets. Additionally, the performance is not very good since there is no clear boundary between classes. On the other hand, Škvára et al. (2018) have shown that kNN is on bar with modern ML tools both in terms of performance and computation times.

DT is a classification algorithm that uses a tree-like structure to establish a relationship between features and the health of the system. Mariniello et al. (2020) developed a damage detection and localization algorithm using DTs on vibration data. Sun et al. (2007) and Amarnath et al. (2013) exploited DTs for rotary machines and bearings, respectively.

BC is a probabilistic classifier that uses conditional probability between features. BC is used by Yu

et al. (2018) to detect the health state of a gearbox, by He et al. (2014) to detect the fault in steel plates, by Muralidharan and Sugumaran (2012) to diagnose pumps, finally by Sharma et al. (2015) to analyze the damage state of roller bearings.

2.1.3 Deep Networks

While traditional methods are mature and well-researched, they are susceptible to long training time, especially when the dataset is large. The advancement in sensing technology and internet-of-things devices increased the volume of the data significantly such that traditional methods may be incompetent to be effective. Compared to traditional methods, deep networks are flexible for learning non-linear relationships between the input (time, frequency, or time-frequency domain features) and output (health state of the structure). The optimization algorithms such as batch gradient descent allow fast convergence to minimize the error between predicted and actual damage class. There are four different approaches for deep networks: i) Auto-encoders (AE) ii) convolutional neural network (CNN) iii) deep belief networks (DBN), and iv) residual networks (ResNET).

AE learns a reduced-order representation of the actual input through encode-decoder structure in a semi-supervised manner (Rumelhart et al., 1985). AE is expected to learn the latent features of the undamaged system such that it can recreate a given input. The error between the input and recreated output constitutes the novelty. If a given input does come from a damaged system, the error will be high indicating the system is damaged. Stacked AE architectures are studied to learn the latent features of frequency-domain data by various researchers (Jia et al., 2016), (Lu et al., 2017), (Liu et al., 2016a) (Xia et al., 2017). Generative AE, specifically, variational AE are also explored by (Ma et al., 2020).

CNN (LeCun et al., 1999) is a commonly-used architecture for damage classification applications. Compared to other deep networks, CNNs allow learning latent features directly from raw time- (Lin et al., 2017), frequency- (Jing et al., 2017), and time-frequency (Islam and Kim, 2019) domain without further processing. Depending on the application, 1D (Avci et al., 2017, 2018) or 2D (Gulgec et al., 2019, 2017) CNN architecture can be used to properly classify the damage.

DBN is also explored towards the detection of novelties. Xie et al. (2018) and Tang et al. (2018) utilized DBN with Nesterov moment (NM) to extract features from rotating machinery and detect bearing damages. Shao et al. (2017) used convolutional DBN (CDBN) with an exponential moving average applied to the learning algorithm, to detect damage of rolling bearings. Guo et al. (2020) studied the applicability of DBN for damage identification of bridges susceptible to noisy and incomplete data.

ResNET gained momentum in recent years since it provides a more versatile architecture through residual blocks and promises better generalization for deep networks (He et al., 2016). ResNet is still a developing

notion within the novelty detection area. Some notable applications focus on the use of time-frequency domain data (Zhao et al., 2017, 2018; Ma et al., 2019).

2.2 Domain Adaptation

Machine learning algorithms, whether it is supervised or unsupervised, assume that the training and testing data come from the same distribution. In time, the test distribution may change which renders the model ineffective. Often, the ML algorithm needs to be retrained for optimal performance. In some situations, obtaining new labeled data for retraining is very costly if not impossible. To remedy this main drawback of ML, a novel approach known as transfer learning gained a lot of interest in the last few decades.

2.2.1 A Brief Discussion on Transfer Learning

Transfer learning covers a broad set of topics (see Figure 2.3). Pan and Yang (2009) categorized the type of transfer learning based on what data is available during training. According to this setting, D is the domain and it is defined by its components; the feature space (X), and the marginal probability distribution, $P(X)$. Given a domain, $D = \{X, P(X)\}$, a task, T is described in terms of the label space, y, and a predictor function, $f(\cdot)$. In this context, for a given task, $T = \{y, f(\cdot)\}$. $f(\cdot)$ is trained on $\{x_i, y_i\}$ where $x_i \in X$ and $y_i \in Y$. For a given x_i in a test data set, $f(\cdot)$ predicts y_i. This predictor function can be also described as $P(Y|X)$.

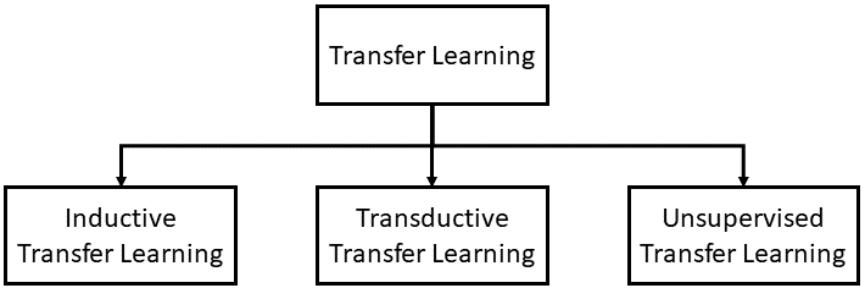


Figure 2.3: An overview on transfer learning - slightly adopted from Pan and Yang (2009)

Pan and Yang (2009) simplifies the definitions further such that the source domain *data* and target domain *data* can be represented as $D_S = \{(x_{S_1}, y_{S_1}, \ldots, x_{S_{n_S}}, y_{S_{n_S}})\}$ and $D_T = \{(x_{T_1}, y_{T_1}, \ldots, x_{T_{n_T}}, y_{T_{n_T}})\}$, respectively. Here, n_S and n_T correspond to the number of samples in source and target domains, respectively.

Transfer learning implies that $D_S \neq D_T$ or $T_S \neq T_T$. When the source and target domains are not matching (i.e. $D_S \neq D_T$), then $X_S \neq X_T$ or $P_S(X) \neq P_T(X)$. Likewise, when the tasks do not match across domains (i.e. $T_S \neq T_T$), then $Y_S \neq Y_T$ or $P_S(Y|X) \neq P_T(Y|X)$. For any case, the aim of transfer learning improving the prediction of target predictor, $f_T(\cdot)$ over D_T using the knowledge D_S and T_S.

For inductive transfer learning, tasks for source and target domains are different, whereas marginal distributions of input features may be the same, though this requirement is not imperative. The main assumption

for inductive learning is that labeled data is available in the target domain. This data is used to induce a predictive model, $f_T(\cdot)$ without retraining (Soares, 2011).

In unsupervised transfer learning, no labeled data is available in source and target domains. The tasks are different but somewhat related, in the sense that we are trying to achieve unsupervised learning. This type of transfer learning usually focuses on clustering and dimension reduction problems.

Lastly, for transductive learning, the tasks for both domains are the same but marginal distributions do not match. In this type of learning, we have access to labeled source data and the target data is not labeled. If only a single domain and a single task are involved in the knowledge transfer, the problem can be reduced to covariate shift (Sugiyama et al., 2007). This statement also implies that the conditional probability of source and target domains does not change, (i.e. $P_S(Y|X) = P_T(Y|X)$) for covariate shift problems. According to Pan and Yang (2009), domain adaptation problems are specific transductive learning cases, where the target domain diverges from the source domain ($D_S \neq D_T$). Redko et al. (2020) generalize domain adaptation to include covariate shift.

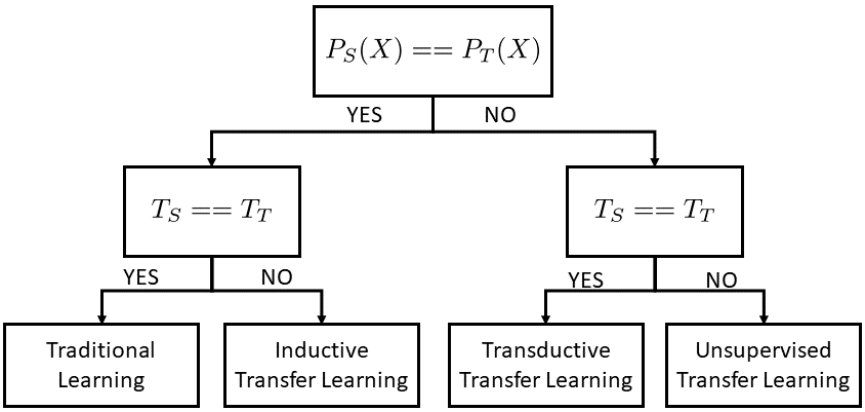


Figure 2.4: Picking the right transfer learning - slightly adopted from Redko et al. (2020)

2.2.2 Conventional Adaptation

Conventional domain adaptation methods simplify the definition of transfer learning by assuming the conditional probability over the source and target domain remain the same ($P_S(Y|X) = P_T(Y|X)$), while the marginal distributions over X are different ($P_S(X) \neq P_T(X)$)

According to the information criteria discussed by Shimodaira (2000), the optimal weights for the transferred model can be obtained by minimizing the expected learning loss by computing the ratio between $P_S(X)$ and $P_T(X)$ as following:

$$\hat{\theta} = \arg\min_{\theta} \left[\frac{1}{n} \sum_{i=1}^{n} \left(\frac{P_S(x_i)}{P_T(x_i)} \right)^{\lambda} l(x_i, y_i, \hat{f}(x_i, \theta)) \right] \tag{2.1}$$

Here, θ is the weights for the model trained with source data; $\hat{\theta}$ is the weights for the transferred model; $\hat{f}(\cdot)$ is the model to be transferred; $l(\cdot)$ is the loss function for the given samples; and λ is a parameter to adjust the probability ratio. This approach is also known as importance reweighting since the empirical loss is reweighted based on the probability of the given input data. Sugiyama et al. (2007) extended this approach by integrating cross-validation error to remove classification bias.

Importance reweighting (see Equation 2.1) requires the estimation of distributions. Another way to match the covariate distributions between source and target domains is obtaining the sample weights using a high-dimensional feature space without computing the empirical distributions. Huang et al. (2006) and Gretton et al. (2009) proposed the Kernel Mean Matching method (KMM) which reweights the source data to match the means of source and target data after mapping them to a Reproducing Kernel Hilbert Space (RKHS). Other methods Gretton et al. (2006) proposed a similar approach called Maximum Mean Discrepancy (MMD) where the objective is finding an RKHS space that minimizes the discrepancy between means of source and target data. This method is regarded as valuable since it is one the of early works that focus on discovering a latent feature representation across both domains. Another well-known method called Transfer Component Analysis (TCA) utilizes MMD to learn *transfer* components (as in *principal* components) to reduce the differences in both source and target domains when projected using these components (Pan et al., 2010). MMD specifically focuses on matching the marginal distributions. When there is a discrepancy in conditional distributions, MMD does not ensure effective transfer. Long et al. (2013) proposed Joint Distribution Adaptation (JDA) to improve the transfer quality further by introducing MMD over conditional distributions. Another notable kernel matching method is investigated by Gong et al. (2012). The so-called Geodesic Flow Kernel (GFK) derives a low-dimensional domain-invariant representation from an infinite-dimensional subspace. Si et al. (2009) proposed transfer subspace learning (TSL) that uses Bregman divergence-based discrepancy as an alternative to MMD. Long et al. (2013) noted that the low-dimensional representation may not be descriptive enough for some domain adaptation problems. Likewise, TSL is remarked as computationally exhaustive, compared to the other domain adaptation methods.

2.2.3 Adaptation for Deep Networks

Many of the conventional approaches discussed in the previous section are adopted for deep neural networks. For example, Deep Domain Confusion by Tzeng et al. (2014) and Deep Adaptation Network by Long et al. (2018) explored the use of MMD within the loss function to find and optimize domain-invariant RKHS. Similarly, Lu et al. (2016) implemented MMD to reveal domain-invariant features towards detecting faults in a gear system working in a variety of loads and speeds. Yan et al. (2017) proposed weighting MMD to treat the class weight bias for deep networks. Other metrics such as Kullback-Leibler divergence (Zhuang et al.,

2015; Lee et al., 2019b), Jensen-Shannon divergence (Zhao et al., 2019), and Wasserstein distance (Lee et al., 2019a) also found their use in deep networks.

Particular domain adaptation applications exploit existing deep network designs to obtain domain-invariant features that generalize well over both source and target domain. For example, autoencoders are known to be an effective tool for dimension reduction and learning latent feature space for a given marginal distribution Kramer (1991a). Zhuang et al. (2015) use autoencoders to learn the latent features that span over both source and target domains and to minimize the distance between distributions. The encoded data is then used to train a label predictor. Another approach is the integration of batch normalization (Ioffe and Szegedy, 2015). Batch normalization is known to be effective in treating covariate shifts. This property is investigated by various researchers to understand its use in domain adaptation problems. Li et al. (2016) proposed adaptive batch normalization to improve the generalization of the network over source and target domain data. A similar approach is pursued by Chang et al. (2019), where domain-specific batch normalization layers are used during the training of the network.

2.2.3.1 Adversarial Training of Deep Networks for Domain Adaptation

In the last five years, with the introduction of adversarial networks, adversarial learning started to replace metrics like MMD in minimizing the discrepancy between source and target domains. Zhang and Gao (2019) categorizes the adversarial domain adaption into three categories: (i) gradient-reversal-based; (ii) minimax optimization-based and; and (iii) generative adversarial net-based.

One of the earliest works that introduced gradient-reversal-based domain adaptation is by Ganin and Lempitsky (2015). In this work, the so-called domain adversarial neural network (DANN) employs a multi-task learning approach and is made of two main components. The first component is the class predictor that extracts features during training and the loss associated with this part is L_C. The second component acts as a domain discriminator and predicts whether a sample is originating from the source or target domain. The associated loss with this component is L_D. A gradient reversal layer combines these two components. During the forward propagation of the training phase, the network behaves likes a typical neural network. On the other hand, during back propagation, the gradient reversal layer multiplies the gradients of L_D with a small negative value. This forces the network to maximize domain confusion and learn domain invariant features. If a latent representation spanning across both domains is discovered, the domain discriminator should not be able to predict the domain origin (as if all of the samples are coming from the source domain). For some cases, gradient reversing may cause negative transfer since data may have complex multimode structures. To alleviate the negative transfer, Pei et al. (2018) extended the DANN by introducing multiple domain discriminators. In parallel, also Zhang et al. (2018b) proposed a very similar approach.

In general, adversarial domain adaptation aims to minimize the class loss while maximizing the domain loss. Tzeng et al. (2015) presented the domain adaptation as a minimax optimization problem where in addition to domain discriminator and class losses, a domain confusion loss is introduced. Later, Tzeng et al. (2017) proposed a new architecture called adversarial discriminative domain adaptation (ADDA). According to this approach, where a neural network is initially trained on source data to predict the labels. This neural network is composed of an encoder (a feature extractor) and a label classifier. The objective of this network is to minimize class loss. In the next phase, the source encoder is coupled with a target encoder and a discriminator. The objective of this network is to maximize domain loss. The weights of the source encoder are fixed, whereas the target encoder is trained with unlabeled target data. The discriminator is expected *not* to predict the correct domain label. Maximization of domain confusion should force the target encoder to map features shared by the source encoder. During testing, the target encoder is coupled with a label classifier which was initially trained with the source encoder. Lastly, Long et al. (2017) proposed a conditional domain adversarial network (CDAN) which integrates domain-specific feature representation and classifier via a multi-linear map.

There are a number of literature that exploits GAN for domain adaptation. For example, Hoffman et al. (2018) introduced the Cycle-Consistent Adversarial Domain Adaptation model (CyCADA) which conceptualizes and minimizes semantic loss, image and feature level GAN losses, and task loss. Another popular approach in this area is proposed by Bousmalis et al. (2017). In this work, the so-called PixelDA uses a GAN to modify source domain images at pixel level as if they are drawn from the target domain. A domain discriminator is trained with the modified source and real target data to maximize domain confusion such that domain-invariant features can be learned by the GAN. In parallel, a task-specific classifier is trained on a real and regenerated the source domain to predict the correct class for the given input. During testing, whether the input is original source data or modified source data (altered to look like target data), or real target data, the classifier is expected to label correctly. (Taigman et al., 2016) used a similar approach. However, the emphasis is more on generating believable samples for the previously unseen (target) domain.

2.3 Physics-guided Learning

The ML tools for detecting novelties are well researched and mature. The majority of these applications exercise a data-driven black-box approach that utilizes a large volume of experimental data obtained directly from the actual dynamic system. One of the obstacles for such methods is often the availability of sufficient training data Zhang and Sun (2021). More specifically, access to a complete a training dataset covering a wide range of conditions is costly and in some instances impossible without actually damaging the system prior to operation. This problem is a major roadblock in developing efficient data-driven algorithms for diagnostics

of dynamic systems for many applications Sadoughi and Hu (2019).

For cases where training data captured from the field is limited, a data-driven black-box ML model could be trained with simulation data. To compensate for the lack of experimental training data, a representative analytical model can simulate the behavior of the system physics to some degree. While physics-based analytical models are capable of generating extensive training datasets, the resulting ML algorithm should still be evaluated with experimental testing data. Well-established analytical models are capable of simulating the dynamic response of the target system Teughels and De Roeck (2005); Jaishi and Ren (2006). On the other hand, calibrating a large set of parameters for complex systems to achieve accurate physical behavior is often computationally exhaustive and at times infeasible Zhang et al. (2020). Eventually, the analytical representation inherits modeling error to some degree. In this case, it is expected that the ML algorithm will fail to perform efficiently during testing since the simulation training data and experimental testing data are statistically divergent Gardner et al. (2020). To address this drawback of data-driven black-box algorithms, the inference should incorporate domain-specific physical knowledge. The physics-guided learning (PGL) which is essentially a hybrid approach aggregating data-driven inference with physical parameters has the potential to leverage the performance of the condition monitoring further and to bridge the gap between simulation and experimental domains.

In recent years, a number of PGL approaches have been proposed. The PGL literature can be categorized into three mainstreams: (i) physics-guided loss function; (ii) residual modeling; and iii) hybrid ML. A more extensive review of these approaches are surveyed by Willard et al. (2020)

2.3.1 Physics-guided loss function

One of the approaches to make ML physics-guided is incorporating physics constraints into the loss function. The loss function for such systems can be simplified as:

$$\text{Loss} = \text{Loss}_{\text{trn}}(Y_{true}, Y_{prediction}) + \lambda \text{Loss}_{\text{phy}}(Y'_{true}, Y'_{prediction}) \tag{2.2}$$

Here, Loss_{trn} is the training loss, Loss_{phy} is the physics constrained loss, and λ trade-off parameter to weight between training and physics-guided loss. Depending on the problem definition, the training loss can be localizing and quantifying the damage (class loss), or predicting the system response (regression loss). The physics-guided loss often does not require new observations taken from the field. The data relating to this loss can be derived through domain knowledge. The physics-guided loss aims to optimize the ML performance by constraining the learning and regularizing the training.

One of the early implementations for physics-guided learning is performed by Karpatne et al. (2017) for estimating lake temperature for a given depth and time along with other parameters. Here, physics-guided learning is achieved in two ways: (i) A hybrid-physics-data neural network model (f_{HPD}) is created where the so-called *drivers*, D is a set of required parameters to make an observation and a physics-based model generates an output, Y_{phy} for the given D. This output (Y_{phy}) is aggregated with D to form the feature vector, X. The network is trained with an input-output pair, where the input is aggregated feature, X but the output is the true observation, Y taken during the experiment. (ii) In addition to the hybrid neural network, a physics-guided loss is added to the learning loss. To achieve the physics-guided loss, first, an auxiliary function that computes the water density for a given time and depth is defined. According to the imposed physics, the density should increase with increasing depth, but the relationship may be nonlinear. During training, for each instance, the density is computed using the predicted Y for a given time. Then, the difference in water density between a depth point and another subsequent (deeper) depth point is calculated. The density differences that violate the imposed physics are averaged and added as a physics-guided loss.

As an extension to the previous approach, Jia et al. (2019) introduced recurrent neural network for predicting lake temperatures. In addition to density-based physics-guided loss, this network exploits another domain-related loss. This loss is formulated as the absolute mean difference between the total lake energy and the energy fluxes going into or out from the lake for a particular time frame. The total lake energy is computed based on the lake temperature predicted by the architecture. The main advantage of both architectures is that the method can generalize with a small amount of data. On the other hand, the architecture requires real observations to achieve successful training.

Zhang et al. (2020) used a CNN to predict the response of a structure subjected to an earthquake. The physics-based loss is defined as the summation of (i) $L2$ norm between the predicted velocity and the derivative state output using graph-based tensor differentiator; and (ii) $L2$ norm of the equation of motion ($m\ddot{x} + c\dot{x} + kx = 0$) that needs to be satisfied. This method also requires experimental data for training.

Another set of architectures that employs physics-guided loss focuses on intermediate variables. The concept of intermediate variable aims to create physics-informed connections among neurons by encoding the physics-related auxiliary parameters into the neurons during the training. These auxiliary parameters are usually not part of the input-output observation, but can be generated empirically or through analysis. The training loss between expected and predicted intermediate physical parameters constitutes the physics-based loss. For example, Daw et al. (2020) used this approach to predict lake temperatures. To predict drag force on a particle suspension in moving fluids, Muralidhar et al. (2019) used intermediate variables such as pressure and velocity fields, and shear and pressure components of drag force. However, this approach acts more like a surrogate model since the observations are taken from a finite element model.

The aforementioned methods focus specifically on regression problems. A damage classification problem studied by Zhang and Sun (2021) couples a neural network with a finite element model updating method. This network utilizes a physics-guided loss that is defined as the cross-entropy loss between the class predicted by the neural network and the class that is estimated by FEM based on the modal properties extracted from the observation. This method also assumes that the designer has access to some labeled experimental data.

2.3.2 Residual modeling

Another approach for incorporating physics knowledge into machine learning is the use of residual networks (see Figure 2.5). This approach employs a model and a neural network in parallel. For a given input, X the model makes a prediction, Y_{phy} and we compute the modeling error, ϵ between the true value, Y and Y_{phy}. The neural network is trained with ϵ and X to make a prediction on the error, $\hat{\epsilon}$. Here, $\hat{\epsilon}$ is used to minimize the prediction error between Y and Y_{pred}.

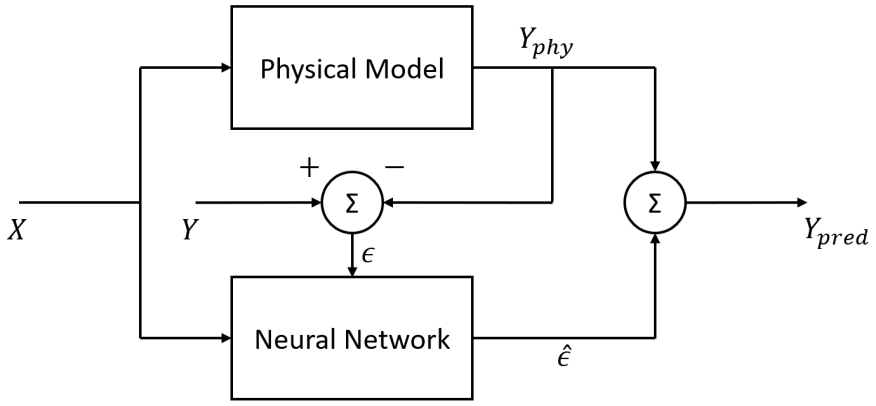


Figure 2.5: Neural network trained with model residuals

The residual models first were discussed by Thompson and Kramer (1994) in the context of modeling chemical processes. The researchers aggregated a radial basis function network (RBFN) with a plant model to estimate the state of penicillin fermentation reaction in time. Another early work by Forssell and Lindskog (1997) focused on using ARX as the model and combine it with a neural network to predict residuals on a water tank system. Xu and Valocchi (2015) used quantile regression forests and support vector regression along with a physics-based model to provide a prediction interval for the flow of a groundwater system. The regressor is trained with the residuals, and it is tasked to predict a bound on the flow based on the distribution of prediction error. Lastly, Liu and Wang (2019) model residuals to train a set of neural networks to predict the progression of heat (heat transfer problem), and temperature and phase for material phase transition.

2.3.3 Hybrid Approaches

More general approaches combine physical models with neural networks that do not involve modeling of residuals. Unlike residual modeling, these methods mostly focus on compensating the deficiencies of physics-based models rather than proposing complete physics-guided learning. For instance, Parish and Duraisamy (2016) used a neural network to augment models that have some deficient physics. They tested this hybrid approach to correct the missing and deficient terms in a turbulent channel flow model. Zhang et al. (2018a) combined a neural network with a problem-specific physics-based proxy-linear model to estimate voltage magnitude and phase for IEEE 57-bus benchmark system. Yao et al. (2018) integrated a physics-based model that is accurate in estimating energy and charge of long-range electrostatic physics with a neural network trained on short-range observations. Similarly, Paolucci et al. (2018) proposed the use of neural networks for short-period range prediction of ground motions whereas a physics-based model predicts the long-period range. Chen et al. (2018) used a neural network to correct the air pollution estimations of a physics-based model that is robust against major factors which evolve over time but not successful in accounting for minor factors.

2.4 Surrogate Modeling

In engineering, the performance of a design is usually validated through a comprehensive analysis. This analysis may involve complex computer simulations as they are known to predict the behavior of physical systems in high-fidelity and to provide a valuable insight about the design process. On the other hand, running high-fidelity simulations is computationally demanding. Especially, in an iterative design process, where broad design space is explored, running simulations repeatedly is exhaustive and extremely time consuming (see Figure 2.6).

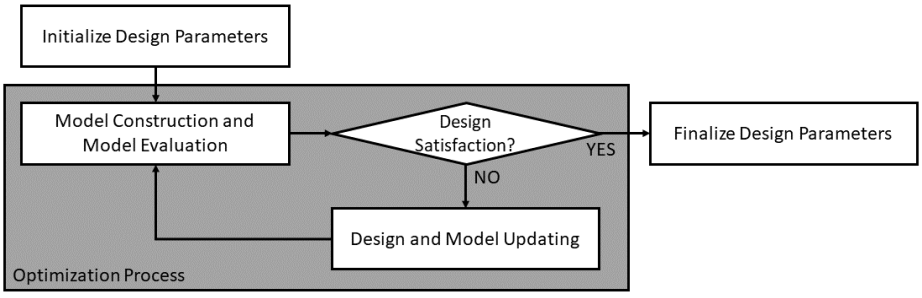


Figure 2.6: Direct optimization using high-fidelity models - loosely adopted from Koziel and Leifsson (2013)

Instead of this cost-prohibitive approach, we can pursue constructing a surrogate model that is capable of approximating the simulation outputs. Naturally, we expect this surrogate model to evaluate given inputs relatively faster compared to a high-fidelity simulation making the design process accelerated and more

affordable.

2.4.1 Designing the experiment and Sampling the design space

A typical challenge associated with constructing accurate surrogate models is finding the best representation that can generalize well across a wide design space. After all, the quality of the surrogate model heavily depends on how the design space is sampled - or how the *sampling void* is filled (Bárkányi et al., 2021).

One naive method of sampling is called polling (Audet et al., 2000). In this mode, a grid space is created across all dimensions for the available design parameters, and the design output space is created accordingly. The vast collection of input and output space is then used for training the surrogate model in a supervised learning way (see Figure 2.7). This type of training is also called one-shot modeling as all samples are collected at once and sampling doesn't involve optimization (Stephens et al., 2011). Instead, all samples are assumed to have the same importance. The size of the grid may impact the overall performance of the surrogate model (Davis et al., 2018). Additionally, the designer is often faced with computational limits and cannot create a dense grid due to a budget. In the event of tight budget constraints, the samples can be randomly drawn from the design space. McKay et al. (2000) proved that tracking the location of the *random* sample and guiding the sampling process may yield more effective design space exploration. In literature, this approach is known as the Latin hypercube sampling algorithm (LHSA). Owen (1992) studied orthogonal arrays and Tang (1993) explored randomized orthogonal array-based LHSA and proposed some optimizations. Loeppky et al. (2009) argue that by using the Gaussian process approach, one can have a general idea about the design space. They proposed an informal rule where the number of runs for an effective experiment design should be ten times the input dimension.

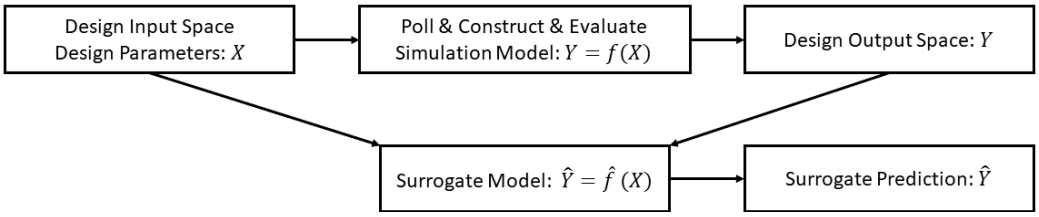


Figure 2.7: Polling from the sample space as in one-shot training

2.4.2 Sequential Design

In general, one-shot experiment design is easy to apply, as it is the most straight-forward way of filling the void when there is no prior information is available regarding the design space. Whether design parameters are sampled using a grid or randomly, the non-linear nature of the model response and the uncertainty of the design space does not warrant a good representation. Furthermore, determining and selecting an optimal sur-

rogate model often requires an exploration-exploitation mechanism depending on the expected outcome (Van Der Herten et al., 2017).

To mitigate the drawbacks of blind polling, we can form our sample space iteratively. This so-called sequential design approach first analyzes the previous and current sampled input-output pairs. Next, the method decides on a new sample from the design space according to a criteria that describes how much difficulty the surrogate model will have to represent the current space (see Figure 2.8). Crombecq (2011) classified sequential design methods into four sections. Those are (i) input-based; (ii) output-based; and (iv) model-based.

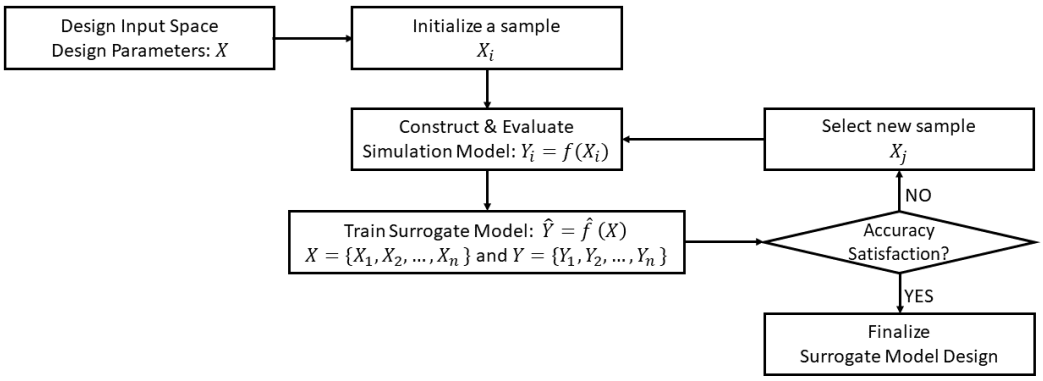


Figure 2.8: Conceptual illustration on sequential design - strongly adopted from Crombecq (2011)

2.4.2.1 Input-based Methods

The input-based methods mainly focus on the exploration of design space and use the information from previous samples to decide on the next sample. One input-sample method is called the low-discrepancy sequence (Hickernell, 1998). Instead of true random sampling, sequences can be generated artificially, where samples are more evenly distributed. If a sequence has no discrepancy, it will be a simple grid. When there is a low discrepancy, some randomization is observed. The convergence of discrepancy is usually defined by the method used. Halton (Halton, 1964) and Sobol (Sobol' et al., 2011) sequences are two of the most popularly used low-discrepancy sequence methods.

According to the definition given above, Latin hypercubes have some sequential design properties, since the randomization is guided to some degree based on previous samples. Qian (2009) proposed nesting subset hypercubes into a Latin hypercube to make it more suitable for some sequential evaluations. Husslage et al. (2005) and Rennen et al. (2010) investigated similar nested hypercube designs towards sequential designs.

Last but not least, other methods such as Voronoi-based design space exploration are studied by Crombecq et al. (2009b). Additionally, Crombecq et al. (2009a) discussed Delaunay triangles as an alternative tessellation for sampling. Finally, Crombecq et al. (2011b) studied a set of Global Monte Carlo methods that use the

distances between samples as a convergence metric to cover most space.

2.4.2.2 Output-based Methods

These methods focus on a balance between exploration/search and exploitation/refinement of the design space based on the sample input and full and/or surrogate model evaluation output values. Crombecq et al. (2011a) proposed a sampling method called LOLA-Voronoi which combines a computationally slow gradient-based local linear approximation (LOLA) algorithm for exploitation with the Voronoi sampling method designed for exploration. They showed that LOLA-Voronoi can satisfy better space-filling properties than Latin hypercubes. Van Der Herten et al. (2014) and van der Herten et al. (2015) integrated fuzzy logic with LOLA-Voronoi to make convergence faster while still having a performance comparable to LOLA-Voronoi.

Osio and Amon (1996) pursued an adaptive sequential sampling using Bayesian-based interpolation. In some cases, Bayesian interpolation may give more weight to some design variables to the point that less significant dimensions are ignored and they are not explored. To mitigate this problem, Farhang-Mehr and Azarm (2005) extended Bayesian interpolation by identifying the *irregular* regions (i.e. regions where the surrogate model may have issues) through a maximum entropy criterion. Kushner (1964) proposed a stable search criterion for noisy outputs. Turner et al. (2007) proposed HyPerSample that provides multiple sampling points per iteration. This approach can be advantageous compared to the methods that account only for one dimension during sample generation. Another Bayesian entropy-based sample method called Sequential Exploratory Experimental Design (SEED) is explored by Lin (2004).

2.4.2.3 Model-based Methods

The methods considered in this topic utilize previous samples, and full and surrogate model responses to provide the next sample. Additionally, the model parameters may be tuned in parallel to make accurate predictions and fewer sample selections. These methods are sometimes also referred to as adaptive sequential sampling-based methods. One main drawback of these methods is that it is specific to the surrogate model used for evaluating the samples.

For example, Gutmann (2001) and Jin et al. (2002) evaluated various sampling methods against RBF-based surrogate models. Use of Kriging-based surrogate models for sample selection optimization based on mean square error (MSE) and maximum entropy (ME) Kleijnen and Van Beers (2004). Liu et al. (2016b) investigated Bayesian surrogate models for sample selection optimization. Busby et al. (2007) proposed an adaptive sampling and model updating method based on local and global prediction errors. Gramacy and Lee (2018) used a similar approach to sample the design space and to update parameters of the surrogate model designed as Bayesian treed Gaussian process. Garbo and German (2017) proposed a method where various

sampling methods are combined with an RBF-based surrogate model selection strategy. They showed that adaptive tuning of the model along with the sample selection reduces the number of samples.

2.4.3 Surrogate modeling from physics-guided learning perspective

The previous section mainly focuses on how to design an experiment and how to sample the design space to create an effective and representative surrogate model. In addition to the literature discussed above, surrogate models based on extended RBF (Zhang et al., 2012), RBF networks (Bajer and Holeňa, 2010), Gaussian process (Liu et al., 2013), smoothing spline (Ratto and Pagano, 2010), and SVM (Lal and Datta, 2018; Shi et al., 2020) explored.

True engineering problems may have highly nonlinear system behavior. As a result, the aforementioned surrogate models may not provide sufficiently high accuracy for high-dimensional problems. To address this observation, new generation approaches started to adopt neural networks as surrogate models. For example, aerodynamic shape optimization is a complex nonlinear problem that involves iterative geometry generation and complicated computation fluid dynamics simulation. Zhang et al. (2021) proposed a neural network-based surrogate model to optimize shapes faster. Another example focuses on predicting railroad bridge displacements while a train is crossing. Understanding the interaction between the bridge and the train usually involves sophisticated finite element models. Han et al. (2019) used a neural network as a surrogate model to predict the bridge displacements. A notable work by Eason and Cremaschi (2014) generalizes the use of neural networks as surrogate models for designing adaptive sequential design.

A set of physics-guided learning problems focus on creating reduced-order models using neural networks to simulate the real system behavior. However, these approaches do not have a particular interest in reducing the number of samples to create a representative design space. Nevertheless, they act as surrogate models. For example, Kani and Elsheikh (2017) used residual recurrent neural networks to reduce the time complexity of nonlinear ODE from $O(n^3)$ to $O(n^2)$ with minimal impact on prediction accuracy. San and Maulik (2018a) and San and Maulik (2018b) used neural networks to predict closure terms in ODE which reduces the time complexity for solving the equations. Lastly, Mohan and Gaitonde (2018) used LSTM to project the high-dimensional dynamics of turbulent flows to a low-dimensional subspace.

CHAPTER 3

Machine Learning based Novelty Detection using Modal Analysis

3.1 Introduction

In recent years, with the emphasis on reliability and sustainability, the interest in Structural Health Monitoring (SHM) has progressively grown. Operations of maintenance, repair, and replacement (MRR) is an integral part of the structure's life-cycle (Rytter, 1993). With the aid of SHM, MRR can be prioritized such that the infrastructure requiring immediate attention can be serviced first. However, due to the presence of environmental and operational variability, it is challenging to develop a reliable damage detection method that informs the performance of the structure accurately (Farrar and Worden, 2007, 2012). Such variations, if overlooked, may lead to incorrect assessment of the structure and cause unnecessary economic loss and social impact. There is still much need for dependable health monitoring approaches that will ensure sustainable civil infrastructure.

Damage detection, also known as *novelty detection*, is, in essence, a method for discriminating significant deviations of a structure from its initial baseline conditions (S., 2007). While ideally the change in the structure can be detected by inspecting features such as natural frequencies, the environmental or operational variations often pollute the baseline and prevent an accurate assessment of the change. Over the last few years, with the advancements in affordable sensor technologies, SHM entered the era of big data Matarazzo et al. (2015); Liang et al. (2016); Wang et al. (2018). As a result of this, machine learning algorithms started to gain traction as a promising damage detection tool for explaining and modeling the relationship between structural responses and integrity under temporally changing conditions while harnessing the power of big data (Worden and Manson, 2006; Farrar and Worden, 2012; Lin et al., 2018).

Damage detection methods employing machine learning can be grouped into two classes: (i) parametric ; and (ii) non-parametric. The parametric approaches often rely on characteristic parameters obtained from structural responses. Such methods often fuse one type of learning algorithms with a preprocessing feature extraction algorithm. For example, system identification can be regarded as a preprocessing algorithm capable of computing features such as natural frequencies, mode shapes, and damping ratios of a structure from raw data. A drastic change of the natural frequency is usually relatable to structural damage. The underlying learning algorithm is expected to capture this damage. Likewise, modal analysis methods, such as cross-correlation functions and frequency response functions can extract other strong features of the structure that provide broader information over time and space (Wirsching et al., 2006). Parametric methods are advan-

tageous over their non-parametric counterparts since they don't need to rely on a numerical model of the structure.

For example, Sohn et al. (2001) developed a parametric novelty detection method that is capable of taking the variations caused by ambient conditions such as a change in loading, temperature, etc. into account to minimize false positive indicators. The method employs AANN to discriminate critical system changes from ambient induced temperature variations. The network is trained via supervised learning to learn the correlation between the variability in the ambient conditions and inherent changes driven by these conditions. The proposed system is tested for a hard-disk model described as a transfer function and it is hypothesized that it could be applied to civil structures. Worden et al. (2003) used a very similar approach involving an auto-associative neural networks (AANNs) and novelty index and evaluated the approach using a more realistic structural system such as a plate supported by stringers similar to a bridge deck. In this study, frequency response functions are used as the input to the network. Novelty detection through machine learning is also investigated for detecting damages of wind turbine blades under fatigue loading. For example, Dervilis et al. (2012a) and Dervilis et al. (2012b) developed a noise tolerant AANN to evaluate the condition of CX-100 wind turbine blade. The frequency response functions were used in this study which is a similar approach to Worden et al. (2003). Zhou et al. (2011) developed two neural networks, one Back Propagation Neural Networks (BPNNs) and one AANN to detect the damage for Ting Kau Bridge in Hong Kong. The BPNN is used to create a correlation model between damage-sensitive modal frequencies and temperature and AANN is employed to characterize the healthy state of the bridge. After the field data is analyzed, an FE model of the bridge is created and simulated to generate new monitoring data where damage was induced in various regions of the model. In addition, the environmental effects are superimposed to the data. Gu et al. (2017) used the modal frequencies of the target structure and the measured temperatures as the input for AANN to improve the generalization capability. In addition, variations in the temperatures causing a change in the frequencies are considered as the input during the training of the network such that false positives can be eliminated. The study looked at the Euclidian distance between measured and estimated frequencies to calculate a novelty index. Their proposed network is tested on a numerical model and in the laboratory on a small-scale test structure. Deraemaeker and Worden (2018) compared the performance of Mahalanobis squared-distance, and Principal Component Analysis (PCA) using real experimental data from a wooden bridge. The features consist of eigenfrequencies and mode shapes measured under changing environmental conditions. Lee et al. (2005) and Mehrjoo et al. (2008) considered a hybrid approach where a finite element model is established as a baseline and neural network is trained to detect the damage based on the expected output of finite element model. These approaches also utilized natural frequencies and mode shapes.

The non-parametric approaches do not require a baseline to establish from structural parameters prior to

deployment and do not depend on the uncertainty of system identification or other modal analysis tools. Non-parametric techniques are advantageous, especially when obtaining a dense array of structural parameters for complex and large systems are challenging. As an example to non-parametric approaches, Abdeljaber et al. (2017) used decentralized 1D convolutional neural networks (CNNs) to eliminate the feature extraction process of typical system identification methods and perform the damage detection directly on the sensor data in real-time. However, sensor data from the healthy and damaged structure is used to train the network for classification purposes which makes the approach supervised learning. Additionally, this study does not consider operational and environmental variability. The algorithm is tested on a grandstand simulator in the lab. In the study, since the trained neural network was not completely successful for classifying the structural condition, specifically producing false negatives, an index reflecting the likelihood of the damage is proposed by computing the ratio of true positives to the total number of test cases. Gulgec et al. (2017) and Yu et al. (2019) used similar approach utilizing deep CNNS to detect damage from sensor data. They also ignore ambient uncertainties. Multiple signal classification (MUSIC) algorithm is another non-parametric approach based on fuzzy wavelet neural networks known to produce successful damage detection from limited sensor data (Jiang and Adeli, 2007; Amezquita-Sanchez and Adeli, 2015; Amezquita-Sanchez et al., 2017).

This chapter introduces an effective damage detection architecture for structures under environmental uncertainty using machine learning. This study utilizes well-established learning algorithms to extract latent features from modal parameters such as natural frequencies and mode shapes under temperature variations and to reconstruct a new representation of these features that is similar, if not identical, to the original. The difference between original and reconstructed parameters constitutes the essential information for detecting critical changes in the system. While modal parameters are known to be a well-researched damage indicator, to authors' best knowledge, this research is the first time that unsupervised machine learning components such as PCA and auto-encoder are applied to utilize mode shapes in addition to natural frequencies for effective damage detection under environmental variability.

As stated above, the approach proposed herein uses the natural frequencies and mode shapes resulting from a well-recognized system identification tool, Natural Excitation Technique and Eigensystem Realization Algorithm (NExT/ERA) as the input and produces a target output which is the expected natural frequencies and/or mode shapes of the system (James et al., 1993, 1995; Caicedo et al., 2001; Brownjohn, 2003; Caicedo et al., 2004). The damage detection relies on the concept of Novelty Index which calculates the mean squared error between input and outputs, e.g. actual and expected natural frequencies, respectively(Deraemaeker and Worden, 2018). To achieve this goal, two unsupervised learning approaches are investigated: (a) principal component analysis (PCA) and (b) Auto-Encoder (AE).

To evaluate and validate the approach along with the learning approaches, a simply supported beam

structure is modeled and simulated in OpenSees under ambient vibration conditions. To add uncertainty to the simulation, temperature, which is known to affect material properties nonlinearly, is varied over a range. The resulting response data constitutes the reference basis for the training data of the machine algorithms. The modal properties of the structure are extracted from this data set, and the machine learning model (model set A) is trained using the aforementioned approaches. In parallel, another set of models (model set B) is developed using only natural frequencies as the input as it is prescribed in previous studies. To demonstrate the advantages of fusing frequencies with mode shapes further, a third model (model set C) utilizing only the mode shapes as the input. Finally, the proposed method is evaluated one more time using the same beam exposed to gradient temperature distribution instead of uniform temperature.

Next, three damage cases are considered where stiffness loss is induced at the midspan at various levels. The structure is again simulated under ambient vibrations, and the resulting modal parameters are fed to the learning model. For the three model sets, novelty index is calculated and the reliability of the results are examined to demonstrate the effectiveness of the proposed approach in detecting the damage.

In addition to the simulations, this study considers a dataset containing laboratory experiments of a scaled three-story structure created by Los Alamos National Labs for further validation. The structure is tested under various damage scenarios simulating section loss at single and multiple columns. An approach identical to the analytical study is used for training the machine learning model and obtaining the novelty index for each case. Lastly, a large-scale three dimensional three-story structure is identified and modeled under temperature gradient. The detection performance of the proposed method is evaluated under multiple damage conditions.

The overall results of the simulations and lab experiments show that the proposed method has, in general, better performance in detecting damage since it utilizes mode shapes as an input in addition to the natural frequencies. In essence, this modal analysis based novel detection approach has the potential to serve as a reliable and near real-time damage detection tool providing accurate data towards objective-driven decisions for maintenance operations. In theory, the end-to-end pipeline considered in this study is capable of streaming real-time data in the time domain from sensors, extract the modal features from the time domain data in near real-time depending on the availability of the computational resources, and compute the novelty index. This approach would indeed accelerate the decision-making process since the state of the target structure is available immediately (Abdeljaber et al., 2017).

In summary, the major contributions of this chapter can be summarized as below:

- A new machine learning approach is proposed that relies on natural frequencies and mode shapes.
- This study streamlines the proposed approach into a pipeline aggregating data collection, system identification, and damage detection.

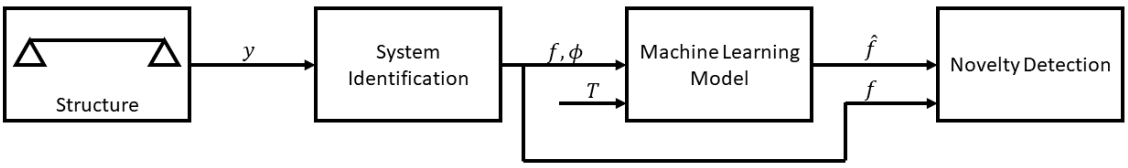


Figure 3.1: Proposed damage detection architecture

- For proof of concept, data from simulation and experimental tests are employed. The performance of the proposed method is evaluated by comparing the detection results to those from prior machine learning methods.
- Results demonstrate that the new approach improves the damage detection rate significantly at the presence of environmental variability.

3.2 Methodology

In this section, the essential components of the proposed approach, illustrated in Figure 3.1, are explained in detail. First, a general description of the target structure and the problems to be solved are described. Secondly, the fundamentals of system identification and feature selection are introduced. The third part of this section focuses on the general architecture of the learning components, and the two machine learning models used in this architecture, PCA and AE. The next part which constitutes the final component discusses the implementation of the novelty detection responsible for determining if the structure is damaged. Finally, evaluation criteria to study the performance of the different learning components in detecting the damage are discussed.

3.2.1 Structure

In this study, we assume the target structure is a system that can be excited with ambient vibration under a variety of environmental conditions. To minimize errors in damage detection, this study omits the structural responses under service loads. As for environmental variations, it has been shown that temperature plays a major role in affecting dynamic features of the system (Woon and Mitchell, 1996; Sohn et al., 1999; Zhou et al., 2011; Abdeljaber et al., 2017). Hence, this study focus on the effect of temperature on material properties.

This chapter investigates three structures. The first one is a finite element model of a simply supported beam. For this beam, a nonlinear relationship between the temperature and the material properties is considered as prescribed by Gu et al. (2017). Additionally, the effect of temperature gradient is investigated to

validate the capability of the proposed method further. The second one is a small-scale three-story structure tested by Figueiredo et al. (2009). Finally, the third one is a three dimensional three-story structure. Section 3.3 presents analytical and experimental structures in detail.

3.2.2 System Identification and Feature Selection

System identification is the process of obtaining dynamic and static characteristics of the structure under service, extreme loads or synthetic excitation. The parameters obtained from identification can be used as an indicator to detect potential damage in the target structure (Doebling et al., 1996). In this study, it is assumed that reliable modal parameters can be extracted from the structure under ambient vibration (Farrar et al., 1996). To minimize the effect of mass change due to the service load and to minimize the errors in damage detection due to the mass change, this study omits the structural responses under service loads. As a result of this, a well-known modal identification method combination, natural excitation technique and eigensystem realization algorithm (NExT-ERA) is used (James et al., 1993; Caicedo, 2011). This method does not require the external excitation acting on the structure and relies on the ambient vibration measurement which is often available on the field. NExT-ERA takes the structural responses to ambient vibration as the input, which are often accelerations measured at specific locations of the structure with sensors. Then the method produces natural frequencies and the mode shapes defining the dynamic characteristics of the structure for that specific measurement instance. While this study focuses on one particular system identification method, any approach that is practically applicable in the field can be adopted.

As mentioned above, this component combines NExT with ERA. Essentially, NExT calculates the free response data from ambient data, whereas ERA extracts natural frequencies, mode shapes, and damping rations from the free response data. Assuming the ambient excitation input is white noise, second order equation of motion can be written as:

$$\boldsymbol{M}\ddot{R}_{\ddot{\boldsymbol{z}},\ddot{z}_i}(\tau)+\boldsymbol{C}\dot{R}_{\ddot{\boldsymbol{z}},\ddot{z}_i}(\tau)+\boldsymbol{K}R_{\ddot{\boldsymbol{z}},\ddot{z}_i}(\tau)=0 \tag{3.1}$$

where $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$ are mass, damping, and stiffness matrices of the system, $R_{\ddot{\boldsymbol{z}},\ddot{z}_i}(\tau)$ is the cross-correlation function between the acceleration, $\ddot{z}_i$ measured at i^{th} location and a reference acceleration $\ddot{\boldsymbol{z}}$. $R_{\ddot{\boldsymbol{z}},\ddot{z}_i}(\tau)$ has the same form as the free vibration response of the structure to be identified. Here, reference signal, $\ddot{\boldsymbol{z}}$ can be chosen as the acceleration of a node on the structure. By computing the cross-spectral density function with respect to the reference acceleration and applying inverse Fourier transformation, the free vibration response

can be obtained in the form of cross-correlation:

$$\ddot{R}_{\ddot{\boldsymbol{z}},\ddot{z}_i} = \frac{1}{N}\sum_{k=0}^{N-1} S_{\ddot{\boldsymbol{z}},\ddot{z}_i}(k) exp\left[j\frac{2\pi kn}{N}\right] \tag{3.2}$$

where $S_{\ddot{\boldsymbol{z}},\ddot{z}_i}(k)$ is the cross-spectral density function of $\ddot{\boldsymbol{z}}$ and $\ddot{z}_i$, k is the frequency index, and n is the time index. More details on NExT are provided by Caicedo (2011) and Caicedo et al. (2004).

ERA utilizes this free vibration data to determine the modal parameters by first constructing the Hankel matrix:

$$\boldsymbol{H}(k-1) = \begin{bmatrix} Y(k) & Y(k+1) & \dots & Y(k+p) \\ Y(k+1) & Y(k+2) & \dots & Y(k+p+1) \\ & & \ddots & \\ Y(k+r) & Y(k+r+1) & \dots & Y(k+p+r) \end{bmatrix} \tag{3.3}$$

where $Y(k)$ is the $m \times n$ response matrix at k^{th} time step. A singular value decomposition on Hankel matrix at $k = 1$, $H(0)$ yields:

$$\boldsymbol{H}(0) = \boldsymbol{R}\boldsymbol{\Sigma}\boldsymbol{S}^T \tag{3.4}$$

where $\boldsymbol{R}$ and $\boldsymbol{S}$ are orthonormal matrices, and where $\boldsymbol{\Sigma}$ is a diagonal matrix containing the singular values. It can be shown that the state-space matrices can be computed as given below:

$$\hat{\boldsymbol{A}} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{R}^T\boldsymbol{H}(1)\boldsymbol{S}\boldsymbol{\Sigma}^{-1/2} \tag{3.5}$$

$$\hat{\boldsymbol{B}} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{S}^T\boldsymbol{E}_m^T \tag{3.6}$$

$$\hat{\boldsymbol{C}} = \boldsymbol{E}_n^T\boldsymbol{R}\boldsymbol{\Sigma}^{-1/2} \tag{3.7}$$

where $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{B}}$, $\hat{\boldsymbol{C}}$ is the estimated state matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, respectively; $\hat{\boldsymbol{D}} = 0$; $\boldsymbol{E}_m^T = [I\,0]$ and $\boldsymbol{E}_n^T = [I\,0]$. Juang and Pappa (1985) discusses ERA method in details.

By applying eigenvalue problem on $\hat{\boldsymbol{A}}$, the modal parameters such as natural frequencies, f and mode shapes, Φ can be calculated by

$$(\hat{\boldsymbol{A}} - \lambda I)\Phi = 0 \tag{3.8}$$

$$w = \left|\frac{\log\lambda}{T_s}\right| \tag{3.9}$$

where $w = 2\pi f_s$ and T_s is the sampling time.

3.2.3 Machine Learning Model

Often the system identification methods are sensitive to the changes in the system induced by damage or environmental and operational effects. However, it is also a challenging task to differentiate the damage from such variations since the baseline is polluted (Sohn et al., 2001). For instance, studies conducted by Ni et al. (2005), Liu and DeWolf (2007), Xu et al. (2009), Xia et al. (2012), Gonzalez (2014), and Li (2014) have shown that the temperature can cause significant changes in dynamic properties of structures. With the aid of the unsupervised learning approaches, a higher-fidelity baseline condition of the structure can be extracted from the polluted data set. Here, the objective of the learning model component is to learn a representation of the data set typically through dimension reduction and to reconstruct a new representation that is similar, if not identical, to the original data. In essence, both principal component analysis (PCA) and Auto-Encoder (AE), also known as auto-associative neural networks (AANNs) can be used to form this behavior. This study uses those two models interchangeably to extract a latent features and evaluates the performance of the architecture by how well the damage is detected. Here, both approaches (PCA and AE) assume that the training data for the learning enabled component contains majorly normal data (from undamaged structure) and very few anomalies (outliers due to instantaneous abnormal events, poor data processing, etc.) (Chalapathy and Chawla, 2019; Chandola et al., 2009). If there is statistically significant event (caused by damage but not environmental effects) deviating from baseline, then unsupervised approaches are expected to capture this event; thus, the error between actual data and the reconstructed/expected data increases. This process is advantageous especially where human experts have difficulty detecting and observing the damage by looking at the data if there is too much variability. The Novelty Index presented here is not a damage classification, but rather a signal that something has changed in the system and owner of the structure may act on this signal considering the risk, operation and maintenance cost.

The learning component is essentially a mapping process and it can be formalized as:

$$\hat{X} = G(X) \tag{3.10}$$

subjected to:

$$min \left\| \hat{X} - X' \right\| \tag{3.11}$$

where X is the input, X' is the subset of X to be reconstructed, and $\hat{X}$ is the output representing the reconstructed X'. $G(\cdot)$ is the mapping function and $\|\cdot\|$ is the normalization operator. To achieve this objective, the mapping function $G(\cdot)$ should be trained with known input X. The input is defined as a set of n natural frequencies where $f = [f_1\ f_2\ \ldots f_n]$ and n mode shape vectors, $\Phi = [\Phi_1\ \Phi_2\ \ldots \Phi_n]$. Ambient temperature,

T taken during the time of measurement can be also added, if available, to the input since it is considered as a feature in damage detection (Zhou et al., 2011; Gu et al., 2017). A complete input from one measurement instance can be defined as:

$$X^{(i)} = vec([T^{(i)}\ f^{(i)}\ \Phi^{(i)}]) \tag{3.12}$$

where i is the index for i^{th} measurement and $vec(\cdot)$ is the vectorization operator. $f^{(i)}$ and $\Phi^{(i)}$ are obtained through system identification and $T^{(i)}$ is the temperature taken during the system identification measurement. Similarly, the input to be reconstructed and the output are defined as:

$$X'^{(i)} = vec(f^{(i)}) \tag{3.13}$$

$$\hat{X}^{(i)} = vec(\hat{f}^{(i)}) \tag{3.14}$$

where $\hat{f}^{(i)}$ is the reconstructed representation of the input, $f^{(i)}$.

Compared to previous research relying only on natural frequencies (Zhou et al., 2011; Gu et al., 2017), this study considers mode shapes also as a valid input. The modal parameters can be obtained from eigenvalue analysis of K and M as follows:

$$[\boldsymbol{K} - (2\pi f_i)^2 \boldsymbol{M}]\{\Phi_i\} = 0 \tag{3.15}$$

When the Young's modulus property of the material, E changes due to the temperature variations, the stiffness matrix, $\boldsymbol{K}$ is affected linearly while $\boldsymbol{M}$ remains same. The relationship between the reference stiffness, $\boldsymbol{K}$ and the temperature-affected stiffness, $\boldsymbol{K'}$ can be simply described by:

$$\boldsymbol{K'} = c\boldsymbol{K} \tag{3.16}$$

where c is a factor defining the linear relationship. When another eigenvalue analysis is applied to K and M, it is observed that the mode shapes stay the same while natural frequencies change as follows:

$$[\boldsymbol{K'} - (2\pi f_i')^2 \boldsymbol{M}]\{\Phi_i\} = 0 \tag{3.17}$$

As an illustration, a two-story shear frame structure with lumped masses and rigid beams studied by Kim

et al. (2012) and Li et al. (2017) is considered as given in Eq. 3.18

$$\boldsymbol{M} = \begin{bmatrix} 2.701 & 0 \\ 0 & 2.701 \end{bmatrix} N/(mm/s^2)$$
$$\boldsymbol{K} = \begin{bmatrix} 558.343 & -279.171 \\ -279.171 & 279.171 \end{bmatrix} N/mm \tag{3.18}$$

An eigenvalue decomposition on this system using Eq. 3.15 will yield the natural frequencies of $f = [1.00, 2.62]$ Hz and mode shapes such that:

$$\Phi = \begin{bmatrix} -0.32 & -0.52 \\ -0.52 & 0.32 \end{bmatrix} \tag{3.19}$$

Assuming c is 1.05, i.e. there is a 5 percent deviation in the stiffness due to temperature, the new stiffness matrix will be:

$$\boldsymbol{K'} = c\boldsymbol{K} = \begin{bmatrix} 586.260 & -293.129 \\ -293.129 & 293.129 \end{bmatrix} N/mm \tag{3.20}$$

Using eigenvalue decomposition presented in Eq. 3.17, the natural frequencies of the *shifted* system will be $f' = [1.02, 2.68]$ Hz. However, the mode shapes will remain unchanged and will be equal to Eq. 3.19. This observation indicates that the mode shapes are independent of temperature variations and should always remain the same as long as the structure is not damaged or the mass of the structure does not change. To sum up, the training algorithm considers the persistence of mode shapes as a statistically important feature for developing a proper mapping function, $G(\cdot)$. The significance of this observation will be discussed further in section 3.2.4.

3.2.3.1 Reconstruction using Principal Component Analysis

Principal Component Analysis is a machine learning algorithm that reduces the dimensionality of a data set leading to a simpler representation of it while preserving essential information that defines the data set (Fukunaga and Koontz, 1970; Goodfellow et al., 2016a). This property of PCA is achieved by computing a linear transformation matrix which can project the original data containing correlated variables to another representation with uncorrelated variables. One main advantage of this decorrelation is exposing the so-called principal components that explain the dominant patterns in the data (Zang and Imregun, 2001; Yu et al., 2010; Tibaduiza et al., 2012). By selecting the prevailing components, one can compress the data, in other words, reduce the dimension of the data, and expose the most important features that are still faithful to the original.

The linear transformation of the PCA can be represented by:

$$Y = X'W_R \tag{3.21}$$

where X' is the $n \times p$ input data matrix, and n rows and p columns correspond to data points (number of measurement instances containing modal parameters) and features (number of modal parameters), respectively. W_R is the $p \times k$ transformation matrix where k is the number of PCA components to be used that explains the majority of variance for the input data. Y is the PCA projection, i.e. a reduced representation of X' with the dimension of $n \times k$. A reconstructed representation of the original input, $\hat{X}$ can be obtained by mapping Y back to p dimensions using W_R^T as following:

$$\hat{X} = YW_R^T = X'W_RW_R^T \tag{3.22}$$

The transformation matrix, W_R is the reduced form of W which is derived through a singular value decomposition on X' such that $X' = U\Sigma W^T$. Here, W is $p \times p$ square matrix and W_R contains the first k singular values of W. Goodfellow et al. (2016a) discusses the derivation of PCA in details.

As an alternative to PCA, nonlinear PCA (NLPCA) proposed by Kramer (1991b) can be also adopted within this architecture since the environmental variations are defined as nonlinear. While this study considers a nonlinear relationship between temperature and structural dynamic parameters, the results from PCA were satisfactory enough not to pursue this adoption.

3.2.3.2 Auto-Encoder

Auto-encoder is a special type of neural network that is trained to reproduce its input as its output (Goodfellow et al., 2016a). Usually, auto-encoder consists of an encoder and decoder, see Figure 3.2. Both encoder and decoder are a set of neural network layers.

Here, the encoder takes the input, X and translates it to H using the mapping function, F described with a set of hidden neural network layers. This mapping can be described as following:

$$H = F(X) \tag{3.23}$$

The encoder extracts the latent representation of the input that captures the most important features.The

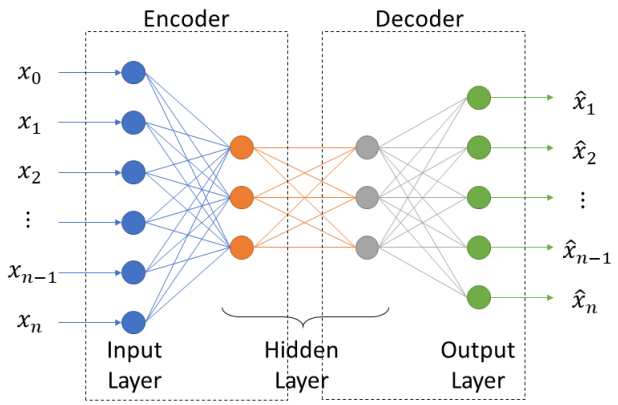


Figure 3.2: An example representation of auto-encoder

decoder function takes H and translates it to $\hat{X}$ using the demapping function, G:

$$\hat{X} = G(H) \tag{3.24}$$

The decoder reconstructs a copy of the input by using the latent representation generated by encoder. In summary, the entire auto-encoder can be rewritten as:

$$\hat{X} = G(F(X)) \tag{3.25}$$

The training process is performed with an objective function to minimize the error between X and $\hat{X}$ given in Eq. 3.11.

It should be noted that Figure 3.2 is an example representation. It is difficult to relate the depth of the network and number of neurons at each layer to physical features such as the natural frequencies and mode shapes (Shwartz-Ziv and Tishby, 2017). By rule of thumb, the network parameters are configured such that the resulting model is generalizable and provides an accurate prediction for untrained data as well (Goodfellow et al., 2016a). As a result of this, the number of layers and the number of neurons should be tuned depending on the complexity of the system. In addition, the auto-encoder can be used to reconstruct the entire input data set or parts of it. For problems where the relationship between the effects of environmental variations is highly nonlinear, AE with nonlinear activation functions is expected to yield more accurate predictions compared to PCA. However, for the examples presented in this study, both PCA and AE provide comparable performance.

3.2.4 Novelty Detection

The objective of the learning component is recovering an *expected* reconstruction of the original input while eliminating environmental effects. To train the learning component and develop a proper mapping function, the training set is expected to be sampled from the measurements while the structure is undamaged. After training, when the approach is given data samples from the undamaged structure, the learning component is expected to create a copy of the input as the output. When damage is present, the mapping will generate faulty copies since the new data set is outside of the training data cloud. The novelty detection component quantifies such differences by identifying the existence of new patterns. This component is well researched and often used in past literature (Worden, 1997b,a; Sohn et al., 2001).

The novelty index (*NI*) that describes the similarity between input and the reconstructed copy can be formulated as follows:

$$NI = \left\| \hat{X} - X' \right\| \tag{3.26}$$

This equation is similar to Eq. 3.11 in nature. Novelty index normalizes the difference between the input (original data), X' and the output (reconstructed data), $\hat{X}$. Accordingly, assuming the training of the learning component is performed successfully, *NI* is expected to be zero or close to zero since $\hat{X} \approx X'$. At the presence of damage, *NI* increases since the learning algorithm produces inaccurate results for $\hat{X}$.

3.2.5 Evaluation Criteria

To quantify the performance of the approach, a modified version of Euclidean distance of novelty index between damaged and undamaged structure is calculated. This criteria can be described as:

$$D_{ud,d} = \frac{\|NI_{ud} - NI_d\|}{\mu_{NI_{ud}}} \tag{3.27}$$

where NI_{ud} and NI_d are the novelty indices for the undamaged and damaged structure, respectively, and $\mu_{NI_{ud}}$ is the mean value of the novelty index for no damage case. Here, Euclidean distance is normalized such that a more reliable comparison can be made between architectures and damage case. The larger this distance, the easier it is to detect the damage based on the novelty index.

3.3 Evaluation of the Proposed Method

This study uses three sets of data to verify the proposed approach: (i) a finite element model of a simply supported beam; (ii) experimental testing of a small-scale three-story structure; and (iii) testing and simulation of a large-scale three-dimensional three-story structure. This section presents and evaluates the results of the structural damage detection performance.

3.3.1 Software Implementation

The structural responses are obtained from the finite element model or the experimental test setup in undamaged and damaged conditions as accelerations, $\ddot{y}$. The accelerations are recorded for some amount of time and saved in a file for each instance of simulation or experiment. Each of these instances containing accelerations is analyzed using NExT/ERA implemented in MATLAB 2018b (MATLAB, 2018). After natural frequencies and mode shapes are obtained from NExT/ERA, this information is vectorized. If the temperature is recorded for an instance, it is also augmented to the vector. Before the training, the data is standardized using *StandardScaler* from scikit-learn toolbox 0.20.2 (Pedregosa et al., 2011) such that each feature has zero mean and unit variance. All the scaled data are saved in their relevant files, based on the condition of the structure.

Next, the machine learning model is trained using the data from the undamaged condition. About half to two-third of the data is used for training whereas the remaining data is utilized for testing and validation to make sure overfitting is prevented. Both PCA and AE algorithms are implemented in Python 3.6.7 (Rossum, 1995). The PCA model is trained using scikit-learn toolbox 0.20.2 (Pedregosa et al., 2011). By trial and error, an appropriate number of components are selected to explain the variance of the data. The reconstruction is performed by first transforming the input data to reduced data and then applying an inverse transformation which is explained in Equation 3.22. AE is trained using Keras 2.2.4 running on TensorFlow 1.12 (Abadi et al., 2015; Chollet, 2015). By trial and error, a neural network with 4 layers (shown in Figure 3.2) is developed to capture salient features of the data. The output of the AE model is the natural frequencies to be reconstructed. The models that contain the natural frequencies and mode shapes are called model set A. In parallel, another set of models (model set B) is developed using only natural frequencies as the input. Additionally, a third model (model set C) is trained which uses only mode shapes.

Finally, the novelty index is obtained by comparing the input natural frequencies with the output for model set A and Bor by comparing the input natural frequencies with the output for model set C. Effectively, there is one novelty index for each vector. Data from different damage conditions are tested as well in this last step. This step is also implemented in Python.

It is important to note that, specifically for model set A, while mode shapes could also be a part of the output vector to be reconstructed, the scale of mode shapes is not the same as frequencies; thus, their contributions to the novelty indices may not be as dramatic as natural frequencies. Moreover, the results presented in the following sections demonstrate that the proposed architecture is capable of detecting damage without reconstructing mode shapes. Reducing the dimension of the output not only accelerates the learning but also reduces the risk of curse of dimensionality (Hughes, 1968).

3.3.2 Analytical Verification with Simply Supported Beams

A simply supported steel beam used by Gu et al. (2017) with a span length of $L = 5.0\ m$ is discretized into equally long 40 member having a cross-sectional area of $A = 1.624 \times 10^{-3}\ m^2$ and moment of inertia of $I = 1.971 \times 10^{-6}\ m^4$. The beam is modeled using finite element modeling (FEM) tool, Open System for Earthquake Engineering Simulation - OpenSees (McKenna et al., 2010). The members are assumed to be elastic-beam column elements. A nonlinear relationship between material stiffness of the elements, E and temperature, T is described as given below:

$$E = [206.216 - 0.4884T + 0.0044T^2] \times\ 10^9 N/m^2 \tag{3.28}$$

where T is in the unit of Fahrenheit. The mass is adjusted such that the structure has the first natural frequency at nearly 0.49 Hz when the temperature is 15°C (∼60°F).

Following the architecture discussed in the previous section, the training and validation data set for the undamaged structure is developed by applying ambient vibration made of white noise to the supports of the beam vertically. The input white noise has a bandwidth of 1024 Hz and the peak displacement is about 0.1 g. The vertical acceleration responses to the ambient excitation at 39 nodes (excluding 2 support responses) are sampled at 200 Hz for 300 seconds. For each simulation, the ambient temperature governing the material stiffness (see Eq. 3.28) is randomly varied between -15°C and 50°C bounded by a uniform distribution. The temperature range is selected to lay out the nonlinear relationship between temperature, material stiffness and natural frequencies fully (see Figures 3.3 and 3.4). Additionally, the distribution allows the environmental effects to contaminate data over the entire temperature range. Figure 3.3 illustrates the temperature vs. stiffness computed according to Eq. 3.28 for the undamaged case. The difference between the minimum and maximum values of stiffness corresponds to nearly 10 percent of the minimum stiffness. Figure 3.4 demonstrates the temperature vs. identified frequency distribution for the undamaged case. The difference between the minimum and maximum values of natural frequencies corresponds to nearly 4 percent of the first natural frequency of the undamaged structure. A set of damage conditions are defined for this structure (see Table 3.1). In total, 4000 simulations are executed. NExT/ERA is performed on the resulting data to extract the first six natural frequencies, f and mode shapes for each natural frequency, Φ. These first six modes also constitute the features to be used for damage detection in accordance with Gu et al. (2017). From each simulation, including the ambient temperature, six natural frequencies, and 234 mode shape points (6 modes $\times$ 39 mode shape points per mode), a vector of 241 data points is created which establishes the input data for the learning enabled component. Out of 2000 vectors from undamaged case, randomly selected 1000 vectors are used for training the machine learning component. The remaining data is used for the validation.

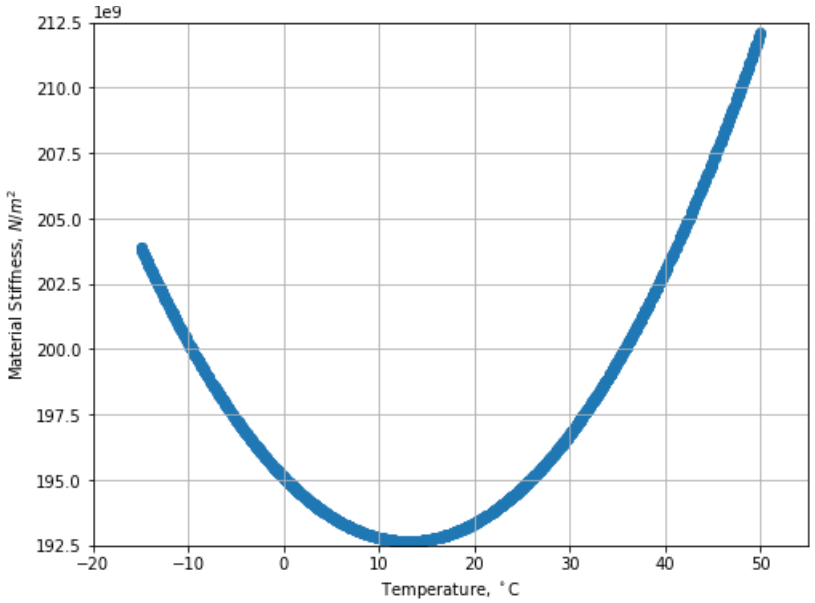


Figure 3.3: Distribution of stiffness with respect to temperature

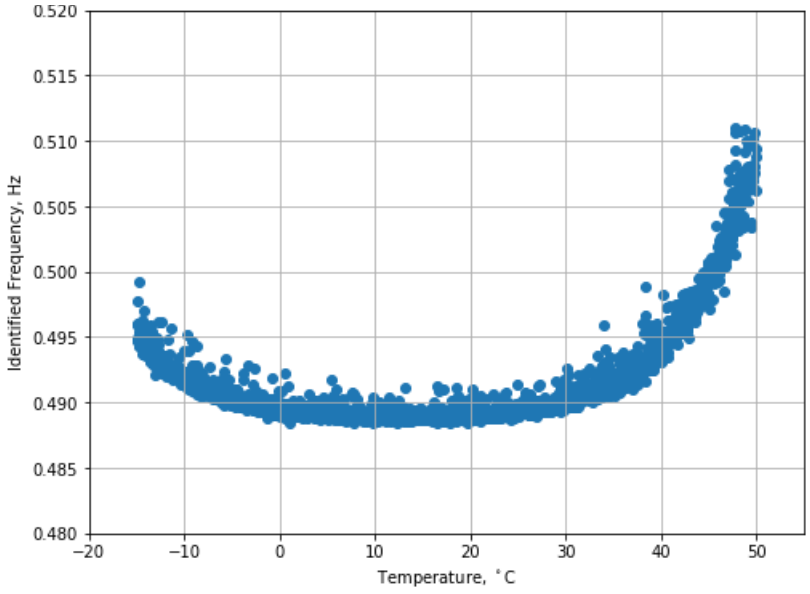


Figure 3.4: Distribution of identified first natural frequency with respect to the temperature

For the three damage cases considered here, the damage is emulated by reducing the stiffness of the 20^{th} el-

Table 3.1: Analytical data matrix

State Condition	Description	No. of Data
No Damage Case	Baseline condition	2000
Damage Case 1	5% stiffness reduction at midspan	1000
Damage Case 2	15% stiffness reduction at midspan	1000
Damage Case 3	50% stiffness reduction at midspan	1000

ement from the left support (corresponding to the midspan) by 5% (Damage Case 1), 15% (Damage Case 2) and 50% (Damage Case 3). For each damage case, 1000 simulation are executed under uniformly distributed random ambient vibrations varying between $-15°$C and $50°$C. It should be noted that the temperature range used in the simulations is rather wide and is not observed for most climate conditions. However, this range also introduces relatively large variability to the natural frequencies. The proposed algorithm is expected to robustly detect damage under large temperature variations.

Figure 3.5 presents the distribution of natural frequencies for the no damage and damage cases using system identification. One can observe that the differences in the frequencies are visually not evident, especially between No Damage Case and Damage Case 1. This can justify machine learning algorithms capable of capturing latent features of the presented data.

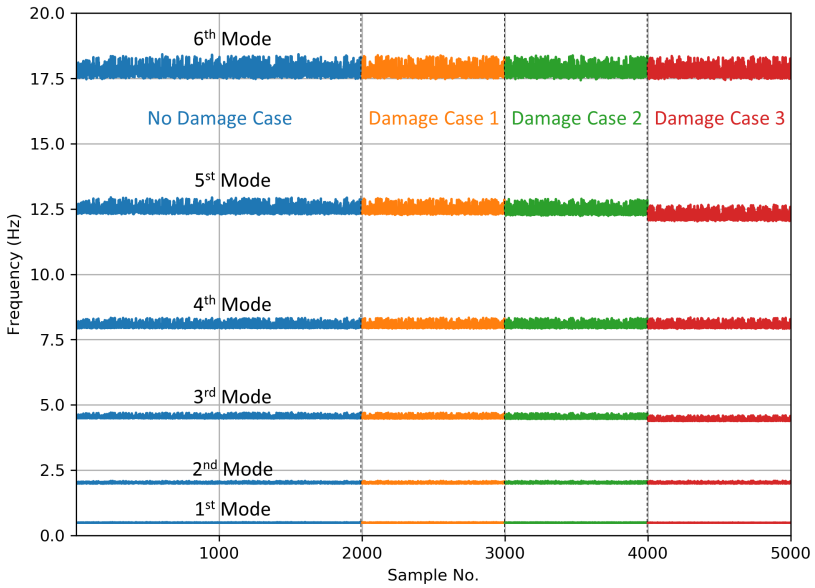


Figure 3.5: Distribution of natural frequencies with varying ambient temperatures for each damage case in analytical data

Regarding the machine learning component, as mentioned before, two architectures are considered: PCA

and AE. For either architecture, the training input is the 1000 vectors each containing the following data points: (i) For model set A, including temperature, 241 data points (1 temperature data + 6 natural frequencies + 234 mode shape data) are packed as a vector from each simulation. (ii) For model set B, only six natural frequencies and temperature data is used. (iii) As for model set C, 234 mode shape points and temperature data are utilized. The number of components used for PCA and network architecture for AE are tabulated in Table 3.2 for each model set.

Table 3.2: Model set properties for analytical data

	Model set A	Model set B	Model set C
Input	1 temperature data 6 natural frequencies 234 mode shape data points	1 temperature data 6 natural frequencies	1 temperature data 234 mode shape data points
Output	6 natural frequencies	6 natural frequencies	234 mode shape data points
PCA component size	100	3	100
AE network structure	241-12-12-6	7-3-3-6	235-50-50-234

The novelty index for both architectures is presented in Figure 3.6. The effect of mode shapes to the performance of the approach is shown by comparing the novelty index of each architecture when mode shapes are used and omitted (model set A, B and C). It is evident from the visual comparisons that including mode shapes into the learning improves the performance of the detection. When mode shapes are not present, there is an overlap between No Damage Case and Damage Case 1 for both architectures. This overlap may lead to false positives or negatives degrading the performance of detection when the damage is small. However, when the damage is larger, the overlap is not observable anymore. To summarize, the proposed approach is successful in capturing the small damage compared to the primitive model which employs only natural frequencies. For large enough damages, the utilization of mode shapes does not improve the outcome of the detection further since the novelty indices are distinguishable enough for primitive models. The modified Euclidean distances computed using Eq. 3.27 for each damage case and architecture are tabulated in Table 3.3. Here, for each case, the novelty indices for the No Damage Case from the validation data set relevant to that case are used as the reference, NI_{ud}. The mean of NI_{ud} establishes $\mu_{NI_{ud}}$. For No Damage Case specifically, the comparison is made between the validation and training data. To calculate $D_{ud,d}$, the complete index vector is used. When the mode shape is introduced to the training, distances become smaller for all cases. For PCA, at the absence of mode shapes, the No Damage and Damage Case 1 values are similar for both architectures. However, AE has a higher distance suggesting that it may be still possible to detect damage with AE. At the presence of mode shapes, the distances are much larger which signifies improved damage detection for the given structure. In summary, evaluation of the architecture performance demonstrates that

the introduction of mode shapes enhances damage detection. When only mode shapes are considered for reconstruction, it is observed that the relative distances increase. This is due to the fact that more features are reconstructed compared to Model Set A and B at the expense of computational complexity. For AE network, reconstructing only mode shapes (model set C) do not improve the detection, whereas for PCA, the sensitivity of model set C is much higher. At this point, it is up to the designer how much sensitivity is desired and what are the computational resources available to reach to the desired damage detection sensitivity. Ideally, an ensemble combining PCA and AE ensures the best detection.

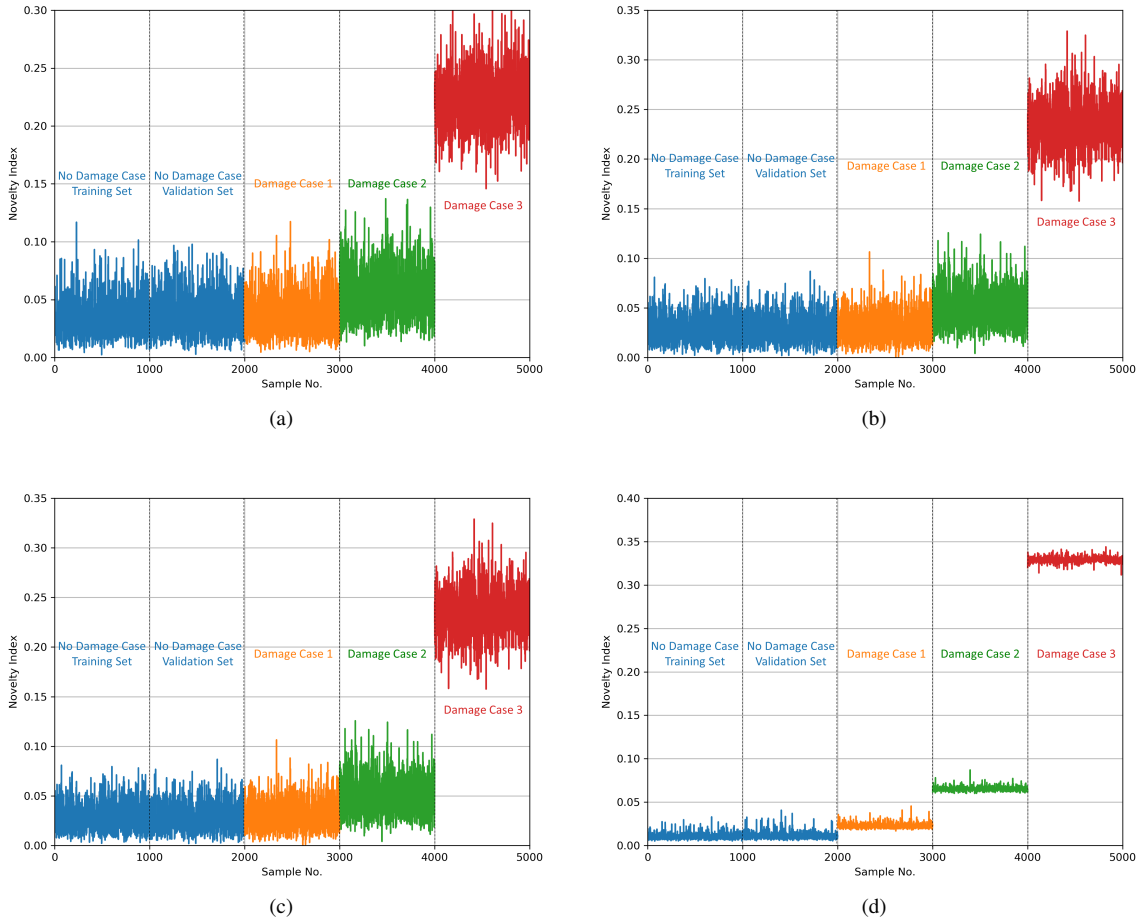


Figure 3.6: Comparison of novelty indices for analytical data: (a) PCA - model set B (mode shapes not included); (b) AE - model set B (mode shapes not included); (c) PCA - model set A (mode shapes included); (d) AE - model set A (mode shapes included); (e) PCA - model set C (only mode shapes included); (f) AE - model set C (only mode shapes included)

3.3.3 Effect of gradient temperature distribution

While in this chapter we hypothesized that the mode shapes do not change under uniform temperature distribution, mode shapes will show slight variation if there is temperature gradient. Such variations may cause

Table 3.3: Modified Euclidean distances for damage cases with and without the inclusion of mode shapes for analytical data

	PCA model set A	PCA model set B	PCA model set C	AE model set A	AE model set B	AE model set C
No Damage Case	21.90	23.30	16.28	22.74	22.89	14.59
Damage Case 0	22.78	38.74	174.30	40.57	24.45	29.32
Damage Case 1	29.50	146.43	637.90	158.40	38.31	129.40
Damage Case 2	168.59	820.53	3333.75	949.47	243.21	770.84

some degradation in the performance of the proposed method. This section focuses on the effectiveness of the method under temperature gradient. Here, it is assumed that the temperature difference between each end of beams is 10°C and changes linearly across the beam. The same number of inputs are used for all the learning components. In addition, to increase the sensitivity of the AE network for model set A, the system structure is modified to 241-50-50-6. The novelty index for both architectures is presented in Figure 3.7. In general, the proposed method can detect the damage under temperature gradient. The overall findings are consistent with the results from the uniform distribution.

3.3.4 Experimental Verification

3.3.4.1 Structure 1

For further verification of the proposed approach, a small-scale three-story structure tested by Figueiredo et al. (2009) at Los Alamos National Laboratory is studied. This structure is excited with an electromagnetic shaker attached to its base. The shaker provided a band-limited white noise and the resulting acceleration responses of the structure is recorded for about 25 seconds at a sampling rate of about 320 Hz (see Figure 3.8). Figueiredo et al. (2009) indicated that the lab environment is not temperature controlled and some temperature variations observed. However, they also did not record the ambient temperature during the experiments. A set of damage conditions are defined for this structure, see Table 3.4. Including no damage condition,

Table 3.4: Experimental data matrix (Structure 1)

State Condition	Description	No. of Data
No Damage Case	Baseline condition	50
Damage Case 1	87.5% stiffness reduction in one first-floor column	49
Damage Case 2	87.5% stiffness reduction in two first-floor columns	50
Damage Case 3	87.5% stiffness reduction in one third-floor columns	50
Damage Case 4	87.5% stiffness reduction in two third-floor columns	45

there are five damage cases for this structure. The damage is introduced by reducing the stiffness of one or two columns at each floor by 87.5 percent. NExT/ERA is applied to all the listed experimental data.

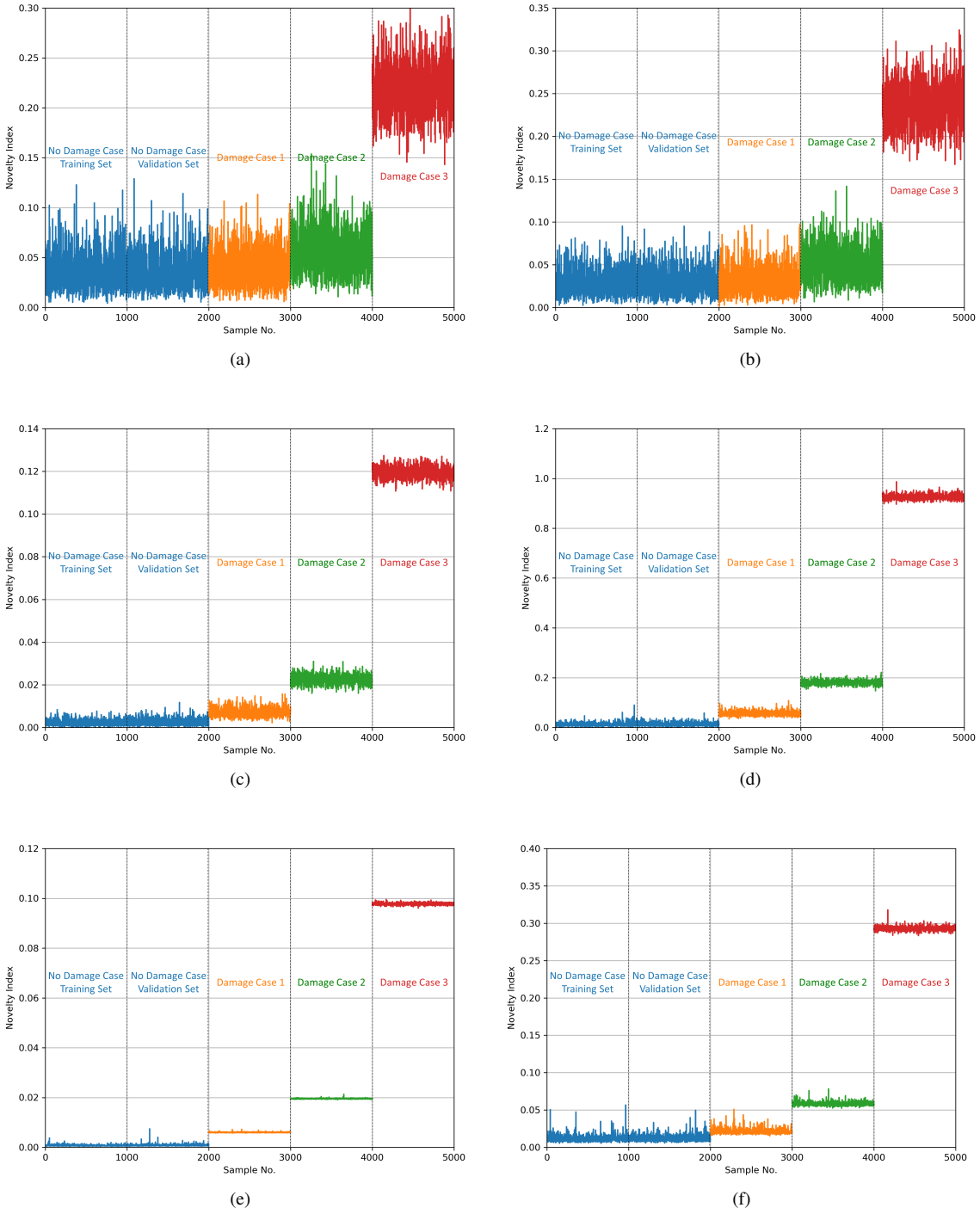


Figure 3.7: Comparison of novelty indices for analytical data under temperature gradient: (a) PCA - model set B (mode shapes not included); (b) AE - model set B (mode shapes not included); (c) PCA - model set A (mode shapes included); (d) AE - model set A (mode shapes included); (e) PCA - model set C (only mode shapes included); (f) AE - model set C (only mode shapes included)

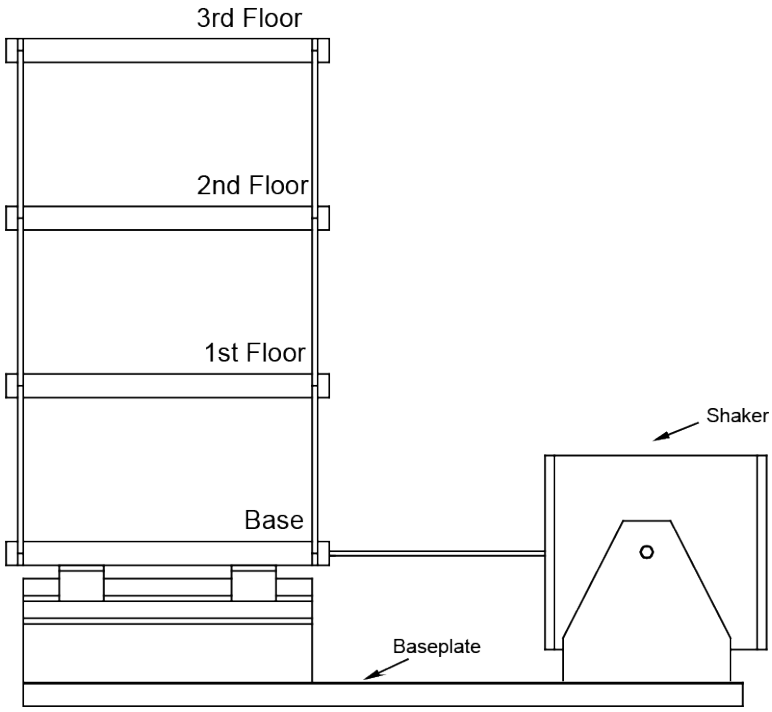


Figure 3.8: Three-story laboratory structure (Structure 1) (Figueiredo et al., 2009)

Since the sampling time is short, the system identification is not able to determine all the dominant modes for all simulations. For the test data, where system identification yields complete natural frequencies and mode shapes are packaged into a vector. Each vector contains three natural frequencies, and 9 mode shape points (3 modes $\times$ 3 mode shape points per mode), summing up to 12 data points. Since temperature was not recorded, this information is excluded in the input. No. of Data in Table 3.4 corresponds to the number of complete data vectors. The distribution of natural frequencies for the no damage and damage cases are shown in Figure 3.9. Similar to the analytical investigation, both PCA and AE architectures are considered. For the training of the machine learning model, the baseline condition is used. Out of 50 data, 40 is used for training and 10 for validation. After the data is standardized, the PCA model is trained with 6 components. The output of the AE model is the three natural frequencies to be reconstructed. A neural network with the dimensions 12-8-8-3 is developed to capture dominant features of the data. Rectified Linear Units (RELU) is used as the activation function on all the layers.

Figure 3.10 presents the novelty index for the experimental data for both architectures with and without the introduction of mode shapes. In general, for all architectures, the damages are distinguished from each other, given the fact that the stiffness degradation was as high as 90 percent. For all figures, when two columns are damaged (Damage Case 2 and 4) the index is higher compared to single column damages (Damage Case 1 and 3). The architectures not relying on mode shapes yield similar indices for Damage Case 2 and 4, whereas

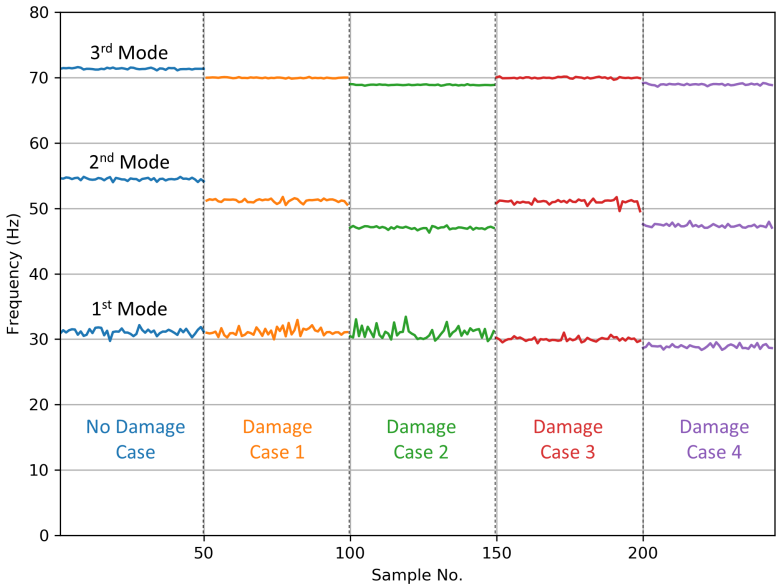


Figure 3.9: Distribution of natural frequencies with varying ambient temperatures for each damage case in experimental data

the utilization of mode shapes as input return distinguishable indices. Considering PCA, some instances of the novelty index for the Damage Case 1 at the absence of mode shapes *leak* to No Damage Case region. This behavior is not observed when mode shapes are introduced. One can notice that with the use of mode shapes, the novelty index of Damage Case 3 and 4 where third columns were damaged is higher for both architectures. The damage at the third floor changes the mode shapes to the point that the novelty index is amplified even though the induced damage is not larger than the first floor. From this observation, it can be concluded that while the proposed approach is a successful damage detection tool, the results may not be definite regarding the magnitude of the damage. To understand the results quantitatively, the modified Euclidean distances for each damage case computed according to Eq. 3.27 are provided in Table 3.5. The validation data set from the No Damage Case is used as the reference. Since the validation data is limited, the data is repeated to match the size of the target. Figure 3.10 implies that when mode shapes are introduced novelty indices may decrease slightly. In parallel, the baseline for no damage case approaches to zero resembling a flat line. Although the novelty indices reduce, the modified Euclidian distance between the baseline and the damage cases increases which implies that damage is quantitatively more distinguishable. Overall, the distances indicate that the presence of mode shapes in the data improves the reliability of the damage detection for the given test structure. The experimental investigation of the data shows that when mode shapes are used in the machine learning models, the damage detection can be more reliable.

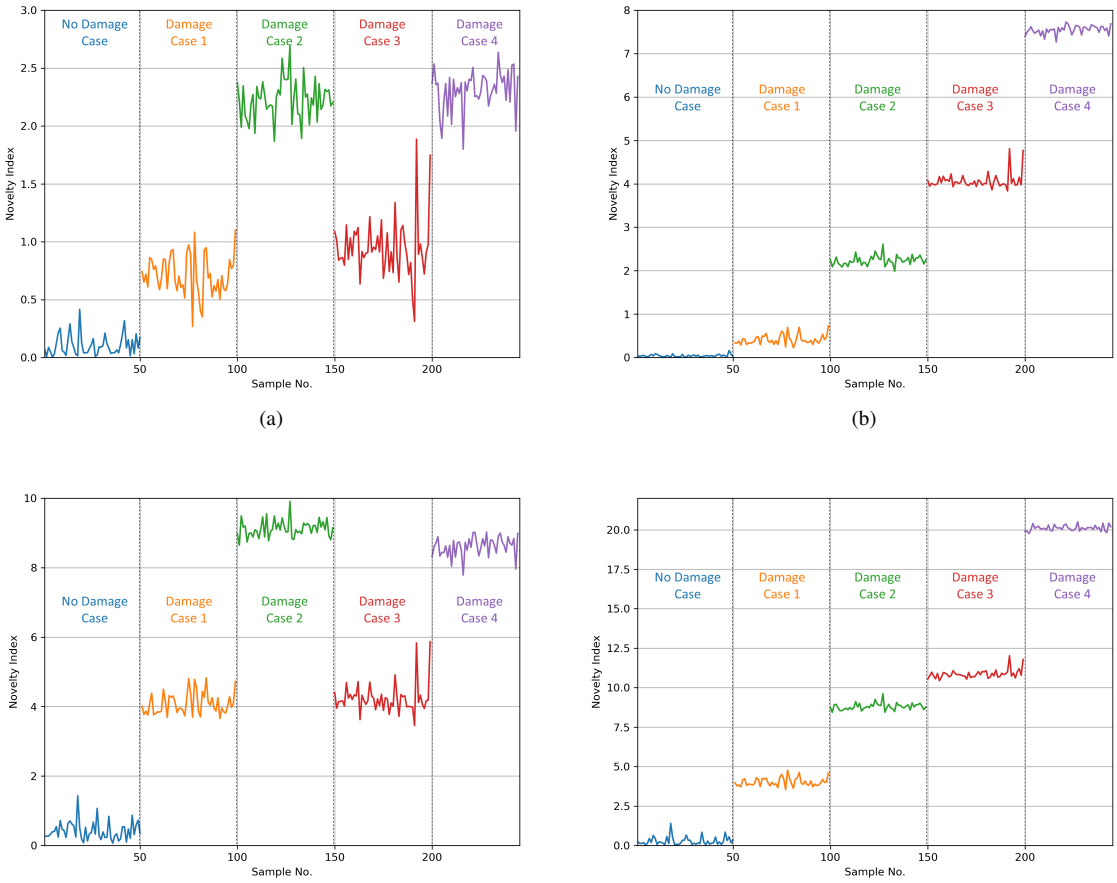


Figure 3.10: Comparison of novelty indices for experimental data: (a) PCA - model set B (mode shapes not included); (b) PCA - model set A (mode shapes included); (c) AE - model set B (mode shapes not included); (d) AE - model set A (mode shapes included)

Table 3.5: Modified Euclidean distances for damage dases with and without the inclusion of mode shapes for experimental data (Structure 1)

	PCA without Mode Shape	PCA with Mode Shape	AE without Mode Shape	AE with Mode Shape
Damage Case 1	29.79	44.92	53.65	76.87
Damage Case 2	104.48	266.75	129.81	177.53
Damage Case 3	42.28	491.04	56.95	220.72
Damage Case 4	105.50	816.04	112.46	395.06

3.3.4.2 Structure 2

This section utilizes a linear three-story three-dimensional frame located at Harbin Institute of Technology (HIT), China (see Figure 3.11 (a)). The prototype structure has a base plan with dimensions 1.84 m by 2.04 m and each story is 1.2 m tall. The structure is braced in one direction with inverted v-brace (see Figure 3.11 (b)). A concrete slab weighting approximately 250 kg is attached to each floor. Including the mass of bare structure, total weight sums to 1066 kg. The columns, beams and girders are made of structural steel with an elastic modulus estimated to be 220 GPa. More details about the system identification and material properties of the structure are discussed in Ozdagli (2015) and Xi (2014).

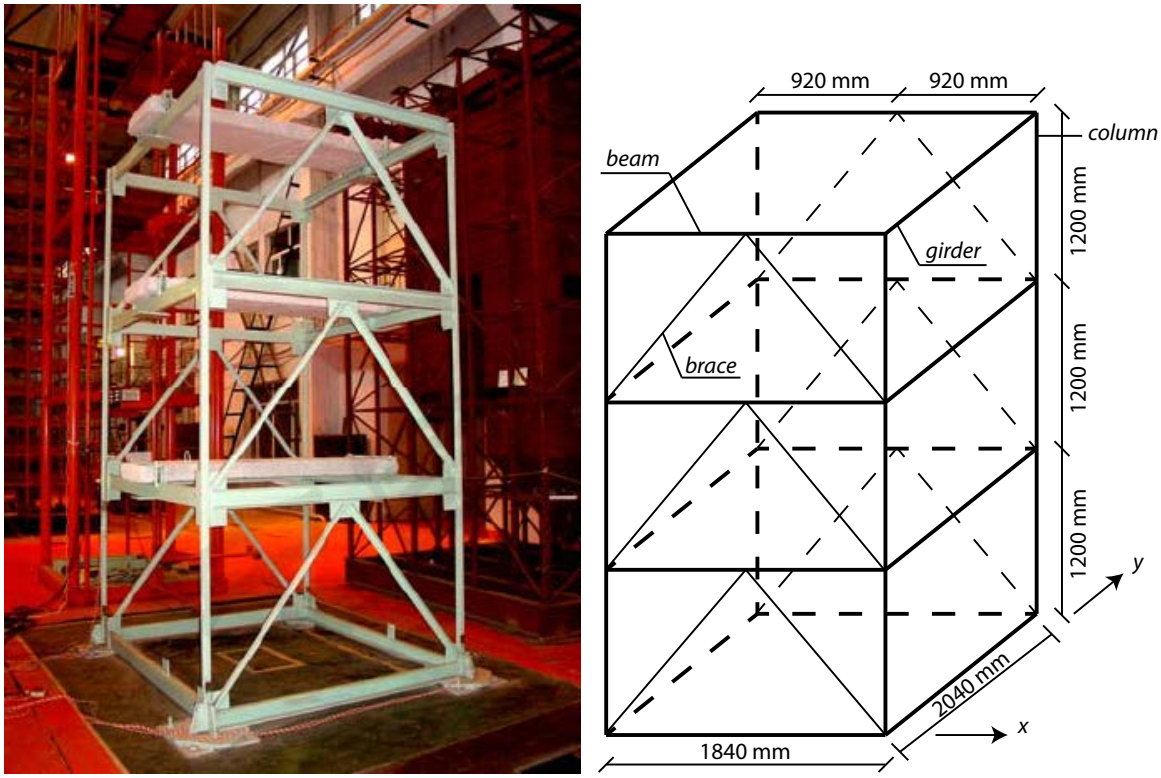


Figure 3.11: Three-story three-dimensional structure (Structure 2): (a) Experimental prototype; (b) idealization

To evaluate the performance of the proposed method, temperature gradient in three dimensions over the structure is modeled. To simulate the temperature gradient, first a finite element (FE) model is established using OpenSees faithful to the experimental structure in terms of boundary conditions and material properties. Each member of the model is criticized into 10 elements, resulting to 360 elements. Equation 3.28 is used to constitute the relationship between the ambient temperature and the material. The model is calibrated to match the experimental modal properties at 15°C. The natural frequencies and mode shapes of the FE model

are presented in Figure 3.12.

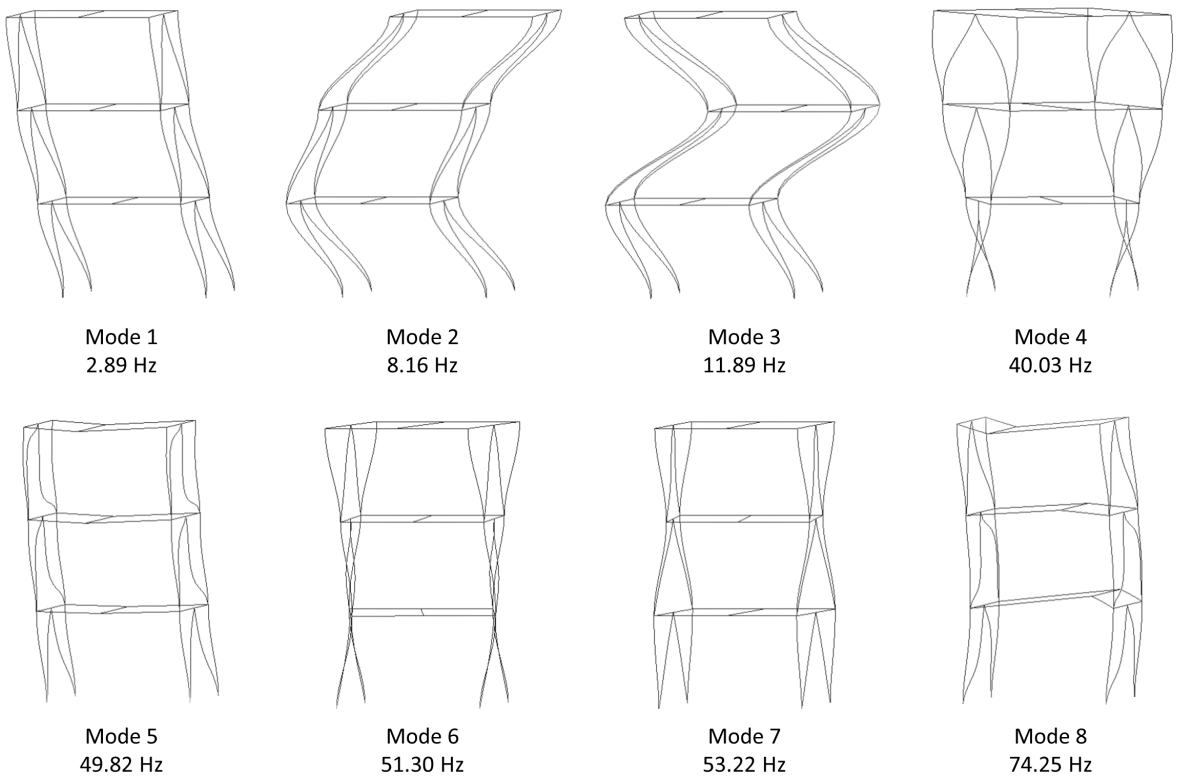


Figure 3.12: Modal properties of the FE model

Four different damage conditions are established for this model with damage state varying between 5 and 10 percent. The damage scenarios are summarized in Table 3.6 and illustrated in Figure 3.13. All damage cases consider a reduction of stiffness only at the midspan element to localize the damage.

Table 3.6: Experimental data matrix (Structure 2)

State Condition	Description	No. of Data
No Damage Case	Baseline condition	2000
Damage Case 0	5% stiffness reduction at midspan of first floor column	1000
Damage Case 1	10% stiffness reduction at midspan of second floor beam	1000
Damage Case 2	10% stiffness reduction at midspan of third floor brace	1000

As for data generation for the validation of the proposed method; it is assumed that temperature gradually increases from bottom node to the top node in the direction of the arrow shown in Figure 3.13. Here, temperature difference at each node relative to the bottom corner of the structure are shown. The maximum temperature difference between upper and bottom part of the structure is 12.5°C. Since it may be challenging to obtain higher order modes in reality, only the first four modes are pursued. It is assumed

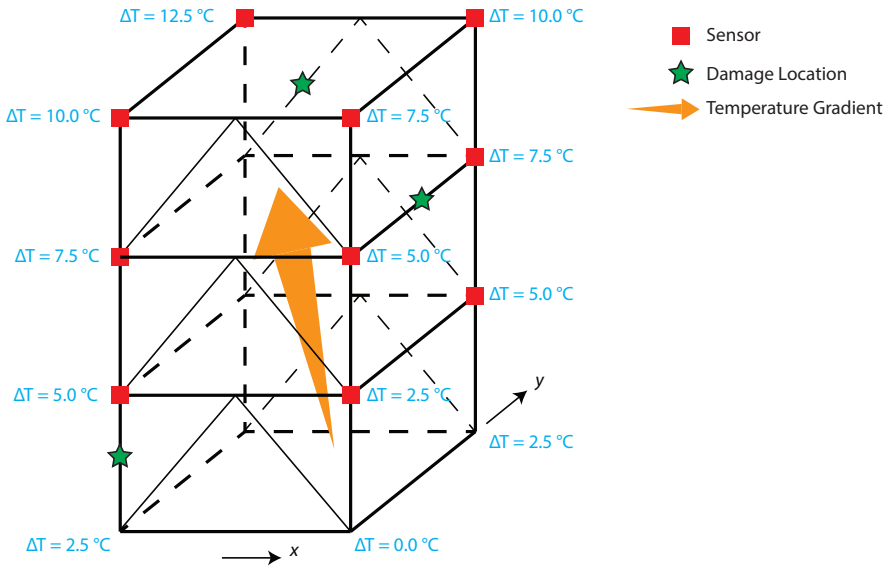


Figure 3.13: Damage conditions and temperature distribution

that accelerometers at each joint (labeled in red color in Figure 3.13) can capture motion along x and y axis which will result to about 24 mode shape points per mode. The method is assumed to have access only to the median temperature. Including the median temperature, four natural frequencies and 96 mode shape point (4 modes $\times$ 24 mode shape points per mode), each input contains 101 data points. For each damage scenario, 1000 inputs are generated whereas for the undamaged case, 2000 inputs (1000 input for training + 1000 input for validation). For all input, the median temperature is uniformly distributed between $-15°$C and $50°$C. Properties of the learning components are summarized in Table 3.7.

Table 3.7: Model set properties for experimental data (Structure 2)

	Model set A	Model set B	Model set C
Input	1 temperature data 4 natural frequencies 96 mode shape data points	1 temperature data 4 natural frequencies	1 temperature data 96 mode shape data points
Output	4 natural frequencies	4 natural frequencies	96 mode shape data points
PCA component size	100	3	100
AE network structure	101-50-50-4	5-3-3-4	97-50-50-96

Figure 3.14 illustrates the novelty indices for all damage cases and model sets. Table 3.8 summarizes the modified Euclidean distance. Accordingly, using both frequencies and mode shapes improves the detection. The performance is more significant for PCA. For this case, reconstructing more features does not improve the detection of PCA when model set A and C are considered. As for AE, combination of frequencies and mode shapes improves the detection for all cases. Here reconstructing more features (model set C) indeed

improve the detection compared to model set A. However, it should be again noted that model set C requires more parameters to train the AE network and predict the detection; thus, computationally more expensive.

Table 3.8: Modified Euclidean distances for damage cases with and without the inclusion of mode shapes for experimental data (Structure 2)

	PCA model set A	PCA model set B	PCA model set C	AE model set A	AE model set B	AE model set C
No Damage Case	27.09	27.48	11.69	0.01	0.01	20.30
Damage Case 1	298919.03	7337.87	124477.62	73.25	30.62	1152.85
Damage Case 2	219040.14	12039.37	20180.22	44.31	32.80	237.35
Damage Case 3	33042.80	2251.73	2352.94	13.88	5.65	122.51

3.4 Conclusions

This chapter proposes a new machine learning architecture to detect damage in structures reliably by incorporating modal properties such as natural frequencies and mode shapes to the training data for the learning components. While the use of natural frequencies in machine learning algorithms is studied thoroughly in the past literature, it has been shown in this study that mode shapes are independent of temperature variations and remain same when the structure is not damaged but material proprieties change due to temperature. As a result of this, the learning algorithm considers the persistence of mode shapes as a statistically important feature. To evaluate and validate the proposed approach, this study uses data sets from a finite element model of a simply supported beam and experimental testing of one small scale and one large scale three-story structure. Both the analytical and experimental investigation presented herein demonstrate that the introduction of mode shapes improves the detection quality significantly in the presence of environmental variability. Especially for detecting a small amount of damages, the performance of the proposed approach is better compared to the architecture which does not utilize mode shapes. Overall, the findings indicate that the proposed approach has the potential to be used as a viable tool in the field. Regarding the practicality of the method discussed herein, there are some considerations to be given. First, NExT/ERA based system identification requires long-time data measurements to capture higher modes successfully. Such data may not be always available. However, the pipeline is flexible enough to allow the use of alternative subspace system identification (SSI) methods such as one proposed by Peeters and Roeck (1999) and known to perform well under noisy environments. Secondly, the success of the method relies on the accuracy of the historical data. A significant change in the system that cannot be identified as damage, such as adding a damper or adding mass will indeed alter the features that were latent in the historical data. In this case, a new model should be trained. Thirdly, temperature gradients along the structure are common during the field measurements. The training data should

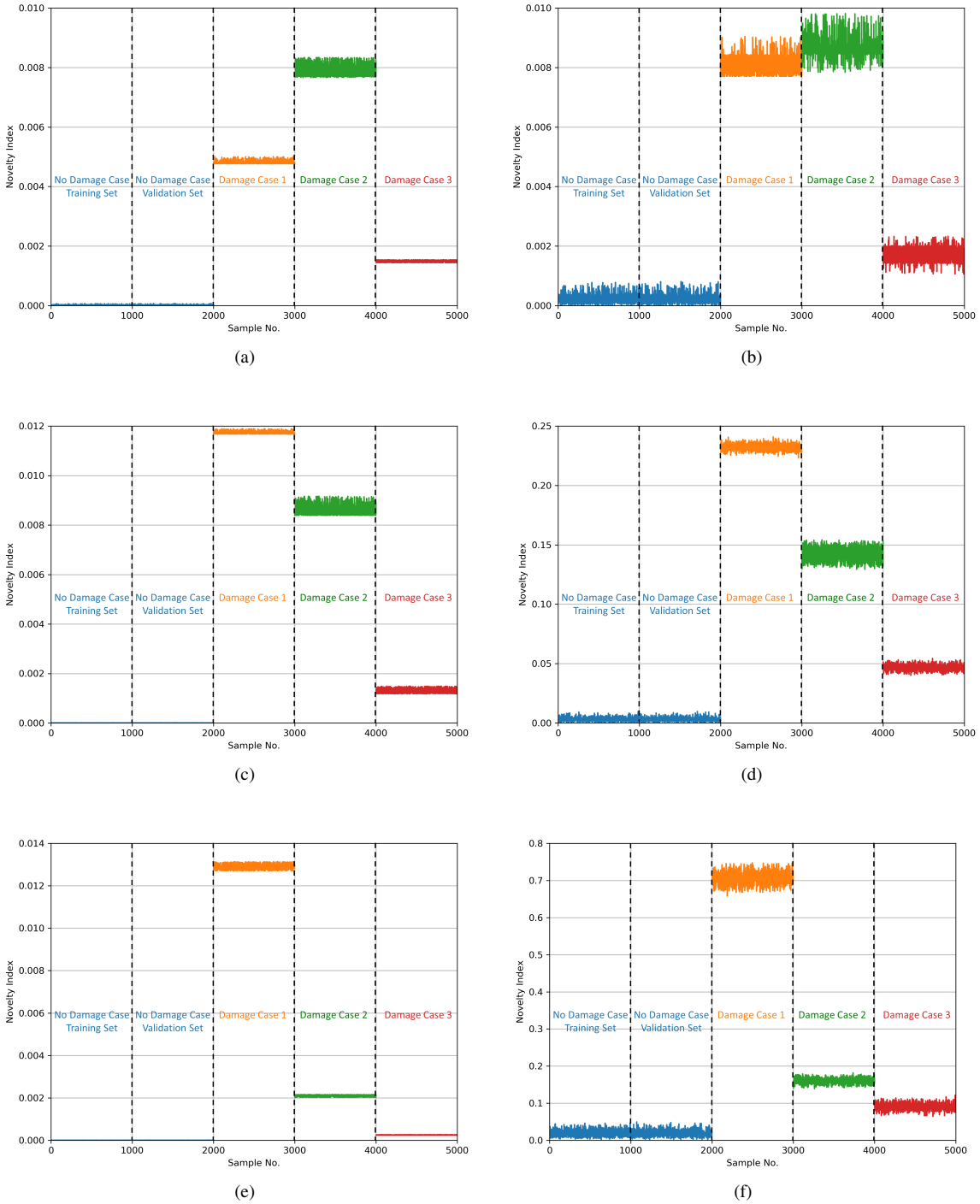


Figure 3.14: Comparison of novelty indices for experimental data (Structure 2): (a) PCA - model set B (mode shapes not included); (b) AE - model set B (mode shapes not included); (c) PCA - model set A (mode shapes included); (d) AE - model set A (mode shapes included); (e) PCA - model set C (only mode shapes included); (f) AE - model set C (only mode shapes included)

consider a wide range of measurements that capture the gradient pattern such that the changes in the mode shapes will not cause novelty indices to increase. Lastly, deep networks inherently require big data for the optimization of hyper-parameters. Extracting features from big data with a huge number of features using PCA can be computationally expensive since the data set is processed as a whole. Either, incremental PCA (Ross et al., 2008), or AE should be used for a system with large sensor arrays. Ideally, an ensemble of PCA and AE architecture should be considered to maximize the detection sensitivity if computational resources are allowing. Future work should include the investigation of the proposed approach in the presence of multiple environmental and operational variability in the laboratory and in the field. Additionally, the data set should be diversified to include not only responses to ambient noise but also forced vibrations due to service loads. Finally, the proposed architecture should be extended to locate damage by relating mode shapes to the spatial data of the bridge under environmental uncertainty.

CHAPTER 4

Domain Adaptation for Structural Fault Detection under Model Uncertainty

4.1 Introduction

United States (US) has one of the most sophisticated infrastructures in the world (World Bank, 2019). However, according to a recent study conducted by the American Society of Civil Engineers (ASCE), the US infrastructure is aging and failure on maintaining it may cost an economical loss in GDP as big as $3.1 trillion (American Society of Civil Engineers, 2013, 2017). The condition of infrastructure for other modern societies is also under stress (Zachariadis, 2018). Overall, it is economically not viable to replace all deteriorating infrastructure due to limited resources, and the operations of maintenance, repair, and replacement should be prioritized accordingly. Acting proactively when a critical infrastructure requires care and preventing catastrophic damages call for novel and innovative approaches.

In the last few decades, structural health monitoring (SHM) has gained a lot of momentum as a means of detecting and localizing damages (Sohn et al., 2002). The introduction of machine learning (ML) into SHM enabled further refinement as mature pattern recognition techniques provide higher accuracy in recognizing structural damages compared to traditional methods (Farrar and Worden, 2012). Among many ML applications, supervised methods are particularly useful (Kiranyaz et al., 2019). Especially, when coupled with artificial neural networks, supervised learning offers promising results for damage detection and localization (Park et al., 2009; Dackermann et al., 2013; Nick et al., 2015).

A majority of supervised SHM applications assume that the data used for training the damage condition classifier has the same distribution as the testing data. However, this assumption is problematic. First, it is unrealistic that one can obtain data belonging to a particular damage condition without actually harming the integrity of the structure before its service (Lu et al., 2016; Gardner et al., 2020). In other words, creating labeled data based on the original state of the structure is not practical for supervised learning models. On the other hand, we can generate a labeled data set using a representative finite-element model or a similar scaled structure where introducing damages is a more cost-effective approach. The collection of labeled normal and damaged state data from this representative structure is called *source domain* and could be used for training a robust damage condition classifier. The second problem with the supervised ML applications is that a model trained with labeled source domain data may fail to predict the condition of the original structure during testing time. The features for the original structure establish the *target domain*. Both source and target domains are distinct in a way that they have probability distributions that diverge from each other. To

summarize, source domain is the model trained on labeled data derived from a representation of the original structure. The model trained on the unlabeled data directly sought from the original structure is the target domain. Both domains have different statistics, which is known as domain shift. The objective of domain adaptation is to design a new learning architecture that generalizes the prediction over both domains (Goodfellow et al., 2016b). This generalization is achieved by finding a mapping that can extract domain-invariant features. Eventually, this mapping is expected to improve the prediction accuracy for the target domain compared to an architecture that does not implement domain adaptation. In brief, transfer of knowledge gained from source domain to target domain is conceptualized as domain adaptation (see Figure 4.1).

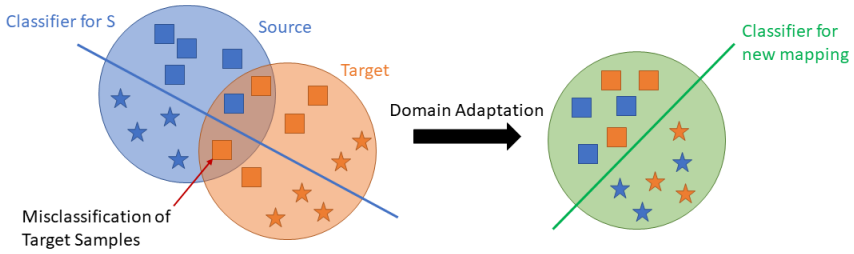


Figure 4.1: Concept of Domain Adaptation

First attempt for domain adaptation started by addressing the distribution shift between labeled training and unlabeled test data. For example, Kernel Mean Matching (KMM) aims to minimize the covariate distribution between two datasets in a higher feature space called Reproducing Kernel Hilbert Space (RKHS) by reweighing the sample data. As a result, KMM is capable of producing a mapping that can match the test data distribution in RKHS (Gretton et al., 2009). While KMM outperforms ordinary classifiers and regressors, the improvement is limited to covariate shift such that the conditional distribution remains same ($P_{train}(y|x) = P_{test}(y|x)$) but input distribution shifts ($P_{train}(x) \neq P_{test}(x)$) across both domains (Bouvier et al., 2019).

Many domain adaptation problems are susceptible to dataset shift where $P(Y|X)$ is not conserved between source and target domains to its highest degree (Wang and Deng, 2018; Wilson and Cook, 2020). Thus, reweighting algorithms are not always effective in such cases. Modern domain adaptation techniques focus on finding a common latent space (also known as domain-invariant feature space) that represents both source and target domains. For example, as an improvement to KMM, maximum mean discrepancy (MMD) metric is introduced to measure the divergence between distributions and to compute a function in RKHS to maximize the difference in expectations between two probability distributions (Borgwardt et al., 2006). A well-known transfer learning method, transfer component analysis (TCA) uses this MMD metric to minimize the maximum expected distribution shift between source and target domain (Pan et al., 2010). Similarly, joint distribution adaptation (JDA) utilizes MMD to measure the statistical difference in marginal and conditional

distributions (Long et al., 2013). Within the SHM community, Lu et al. (2016), Li et al. (2018), and Li et al. (2019) utilized MMD metric as a loss function for the training of neural networks to improve the prediction over target data using both source and target data during training for gear fault diagnosis. Similarly, Xie et al. (2016) and Gardner et al. (2020) applied TCA to classify the damage on gears and structures, respectively.

The new generation domain approaches exploit adversarial training to find domain-invariant features (Wilson and Cook, 2020). These approaches adopt the zero-sum game where a label classifier (the network that predicts the correct label of input whether it is coming from source or target domain) is trained to deceive a domain classifier (another network in parallel that predicts whether the input is source or target domain data). For instance, Domain Adversarial Neural Network (DANN) uses gradient reversal layer during back-propagation to reverse the domain classifier weight derivatives to maximize the domain confusion (Ganin et al., 2016). Adversarial Discriminative Domain Adaptation (ADDA) uses a two-step approach where the network is first pre-trained on source data and then a domain classifier is trained to learn target domain features. As an alternative to DANN-type of domain adaptation, domain mapping approach uses GANs to translate a sample data from target domain to source domain (Benaim and Wolf, 2017; Zhu et al., 2017). However, these applications are limited to image-like domains.

This chapter introduces an effective domain adaptation approach to address the distribution shift between source and target domain for supervised machine-learning-based SHM applications. More specifically, we utilize a domain adversarial neural network (DANN) approach to predict the damage condition of a structure operating under a target domain using both labeled source and unlabeled target domain data during training time. The main purpose of the DANN architecture is learning features that represent both source and target domains. To achieve this goal, DANN implements a multi-task topology that combines a regular feed-forward neural network (NN) based damage classifier using source data with a domain discriminator NN which utilizes source and target domain data. The domain discrimination component enables the feed-forward NN to extract latent features underlying both domains by minimizing H-divergence between domains.

To demonstrate the suitability of the DANN for SHM applications, the chapter investigates two case studies. The first case study focuses on a gearbox system with a set of damage conditions operating under low- and high-load. A DANN model is trained with labeled low-load and unlabeled high-load data to predict the damage condition for the high-load operation of the gearbox. Additionally, for this case, DANN is compared to two well-known transfer knowledge methods, Transfer Component Analysis (TCA) and Joint Distribution Adaptation (JDA) to show the performance gain. In the second case, the effectiveness of the domain adaptation from the numerical model to experimental data is studied for a small-scale three-story structure. The numerical model of the structure is used to simulate various damage conditions for the source domain whereas the experimental data constitutes the target domain. Results from both case studies indicate

that domain adaptation is a viable method for SHM applications, and it increases the accuracy of damage condition prediction considerably. Additionally, the DANN can be considered as a potential ML architecture enabling appropriate knowledge transfer across the source and target domains.

For many machine-learning-based SHM applications focusing on damage detection and localization, a shift from source to the target domain is expected. Domain adaptation is a viable methodology for minimizing the distribution shift between source and target domains. This chapter demonstrates that DANN is a suitable approach for learning latent features that underline both source and target domains. The case studies examined in this chapter show that DANN improves the prediction accuracy of supervised damage detection and localization algorithms.

Overall, the major findings and contributions of this chapter can be summarized as below:

- For many machine learning-based SHM applications focusing on damage detection and localization, a shift from source to the target domain is expected. Domain adaptation is a viable methodology for minimizing the distribution shift between source and target domains.
- This chapter demonstrates that DANN (Ganin et al., 2016) is a suitable approach for learning latent features that underline both source and target domains for SHM applications. The case studies examined in this chapter show that DANN improves the prediction accuracy of supervised damage detection and localization algorithms.
- The effectiveness of DANN is compared to the black-box approach and traditional knowledge transfer methods called TCA and JDA. Our results show that DANN outperforms all three architectures.

The rest of the chapter is outlined as follows. First, Section 2 discusses condition monitoring briefly and formulates the domain shift problem. This section introduces the DANN model for SHM applications as well. Section 3 presents case studies and the evaluation results. Lastly, Section 4 summarizes the paper and draws conclusions.

The code to generate the results in this chapter can be accessed from https://github.com/aliirmak/DASHM.

4.2 Domain Adaptation in SHM

In traditional SHM applications, vibration data is captured from various locations of the structure in the form of accelerations (Abdeljaber et al., 2017; Ozdagli and Koutsoukos, 2019). Meaningful features extracted from these measurements through time or frequency domain analysis establish the input space for a supervised learning model. Each data in the input space can be associated with a label describing the structural condition in terms of the location of the damage and its intensity to form $\{X, Y\}$. Supervised learning algorithms require access to those labeled data for proper training. While the no-damage/normal data is often available when

the structure is first erected, it is impractical to abuse the structure just to obtain the data relevant to various damage conditions.

As a solution to the main fallback of the supervised learning methods, model-based SHM approaches exploit numerical models to establish a baseline for damage detection and damage localization (Mirzaee et al., 2015; Figueiredo et al., 2019). Numerical models can be useful for generating labeled source domain data. However, an ML model trained with source domain data may suffer from the uncertainty gap between the numerical model and the experimental structure (Catbas et al., 2013). Consequently, the learning model may not yield correct labels for the unlabeled target domain and may diagnose the damage improperly for the target structure. From the domain adaptation perspective, the distribution shift between source and target domain should be addressed (Singh et al., 2020; Li et al., 2020). Accordingly, the problem for supervised SHM applications is finding domain-invariant features that represent both labeled source and unlabeled target domain.

In this chapter, the source domain D_S consists of labeled data derived either from numerical simulations or from a particular state of the structure (for example, low wind, low traffic load, low-load, etc. corresponding to the normal operation). The target domain D_T is either the data captured from the experimental structure or an operational state of the structure that is not relevant to source domain (such as high wind, high traffic, high-load, etc. corresponding to stressing operations) and it is unlabeled. Then, the typical domain adaptation task for supervised SHM application is predicting the class for unlabeled target domain data using the knowledge gained from both source and target data.

For SHM, it is natural to consider a classification task where $X = \{x_i\}_{i=1}^{N}$ is the input space of features and $Y = \{y_i\}_{i=1}^{N}$ is the output space corresponding to the labels. Suppose that we have two different distributions over the $\{X, Y\}$: *i)* D_S is the source domain which contains the labeled source samples with $S = \{(x_i, y_i)\}_{i=1}^{n} \sim D_S$; and *ii)* D_T is the target domain which consists of the unlabeled target samples with $T = \{x_j\}_{j=1}^{n'} \sim D_T$. We assume that the distributions for both domains are different such that $D_S \neq D_T$. This implies that the distributions for the input space from S and T are not identical, namely $p(X_S) \neq p(X_T)$. Similarly, the conditional distributions that are used for inference may not match, that is $p(Y_S|X_S) \neq p(Y_T|X_T)$. Given D_S and D_T, the task for the domain adaptation is to build a classification model $h(x)$ which can predict correct labels for samples from D_T using the knowledge learned from D_S and D_T.

4.2.1 Domain Adversarial Neural Network

A common domain adaptation approach is finding a mapping function that can minimize a probabilistic discrepancy metric between the two domains. The majority of these metrics focus on computing the divergence, i.e., the distance between two probability distributions. For example, the kernel mean matching (KMM)

algorithm minimizes the mean distance in a kernel space by re-weighting the target domain with respect to source domain(Huang et al., 2007). The approach in Sugiyama et al. (2008) proposes to minimize the Kullback-Leibler (KL) divergence for minimizing domain shifts. A well-known transfer learning algorithm called transfer component analysis (TCA) utilizes Maximum Mean Discrepancy (MMD) to minimize the distance between two domains in Hilbert space (Sejdinovic et al., 2013; Pan et al., 2010). Lastly, Ben-David et al. (2010) hypothesize that a classifier-induced divergence, namely H-divergence is sufficient for domain adaptation.

H-divergence relies on distinguishing the examples of D_S and D_T and computing the domain divergence from the data in both domains. Accordingly, we label the data from D_S and D_T as *0* and *1*, respectively. Then, we have a new dataset that can be described as:

$$U = \{x_i, 0\}_{i=1}^{n} \cup \{x_j, 1\}_{j=1}^{n'} \tag{4.1}$$

Then, the objective is to develop a function that predicts the class of the sample input χ correctly, i.e., $f : \chi \rightarrow [0,1]$. Similarly, $h'(x)$ is the learned model $h' : \chi \rightarrow [0,1]$. Then, the generalized error is:

$$\varepsilon = E[|h'(x) - f(x)|] \tag{4.2}$$

Given ε, the H-divergence is approximately:

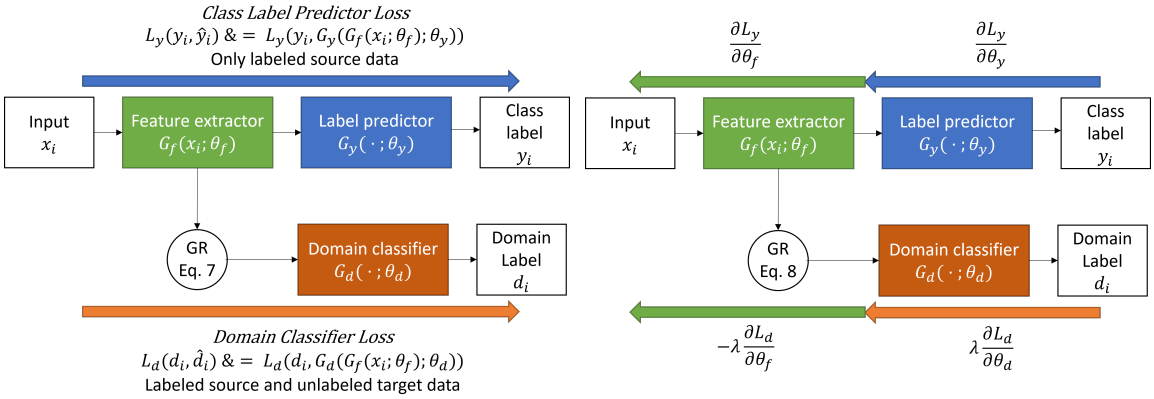
$$d = 2(1 - \varepsilon) \tag{4.3}$$

One purpose of the domain adaptation is minimizing the H-divergence d. More details on the derivation of the divergence can be found in Ben-David et al. (2010); Ganin et al. (2016).

There are inherently two tasks for implementing H-divergence based domain adaptation. First, we want to train a domain classifier $h'(x)$ that can discriminate between source and target domains. At the same time, we want to design a class predictor $h(x)$ to correctly predict the class for the source domain data during training. It should be noted that the class predictor cannot be trained on target domain data as they are unlabeled. The ultimate aim of the domain adaptation is finding features that underline both the source and the target domain. Such a representation is expected to minimize the H-divergence and the domain predictor $h'(x)$ should not be able to distinguish between the source and target domains.

The domain-adversarial neural network (DANN) approach introduced in Ganin et al. (2016) exploits this objective by proposing a multi-task learning approach. The DANN is composed of three components: feature extractor, label predictor, and domain classifier). Figure 2 illustrates the training (forward and backward

propagation) and testing phases. Here, the feature extractor (green colored) is a set of neural network layers that extracts domain-invariant features for a given input. The label predictor (blue colored) layers predict the label of a given input based on the domain-invariant features computed by feature extractor layers. The domain classifier (red colored) is tasked with discriminating the domain of the input whether it is originating from source or target domain. Finally, a gradient reversal layer denoted as GR connects feature extractor to domain classifier. GR changes its behavior based on the type of propagation.



(a) Computation of loss - Forward propagation

(b) Computation of gradients - Backward propagation

(c) Testing phase

Figure 4.2: Simplified DANN architecture

During the forward propagation (see Fig. 2 a), the class label loss is computed only over the labeled source domain data, whereas the domain classifier loss is calculated using both labeled source and unlabeled target domain data. In this phase, GR acts as a linear function and does not modify the propagation of loss. During the backward propagation phase (see Fig. 2 b), the gradients of the losses are computed. In this phase, GR reverses the gradient by multiplying the propagation with a negative small constant. This negative gradient maximizes the domain confusion by enforcing latent features extracted from both source and target domain to be indistinguishable. After training is complete, one can test the network only using feature extractor and label predictor layers.

For a general DANN, the loss function can be formulated as following:

$$\mathscr{L} = \frac{1}{n_s}\sum_{x_i \in D_s} L_y(y_i, \hat{y}_i) - \frac{\lambda}{n_s + n_t}\sum_{x_i \in D_s \cup D_t} L_d(d_i, \hat{d}_i) \tag{4.4}$$

where n_s is the number of source domain sample; n_t is the number of target domain samples; L_y and L_d are the class label and domain label loss, respectively; λ is the trade-off parameter between class label and domain

label loss; x_i is the i_{th} input; y_i and $\hat{y}_i$ are the corresponding true and predicted class labels, respectively; d_i and $\hat{d}_i$ are the corresponding true and predicted domain labels, respectively.

Next, we denote the feature extractor layers as G_f, label predictor as G_y, and domain classifier as G_d. Accordingly, L_y and L_d can be formulated as:

$$L_y(y_i, \hat{y}_i) = L_y(y_i, G_y(G_f(x_i; \theta_f); \theta_y)) \tag{4.5}$$

$$L_d(d_i, \hat{d}_i) = L_d(d_i, G_d(R(G_f(x_i; \theta_f)); \theta_d)) \tag{4.6}$$

where θ_f, θ_y, and θ_d are the weights for feature extractor, label predictor, and domain classifier, respectively. Here, $R(x)$ represents the gradient reversal layer as a function. The GR function has distinct behavior for forward (see Eq. 6.4) and backward propagation (see Eq. 6.5) as prescribed below:

$$R(x) = x \tag{4.7}$$

$$\frac{dR}{dx} = -\mathbf{I} \tag{4.8}$$

The computed loss and gradients based on the GR function, are given in Figures 4.2a and 4.2b. Then, the objective of the training is finding the weights, θ_f, θ_y, and θ_d that optimize the joint loss, $\mathscr{L}$ given in Eq 5.3 by minimizing the label predictor loss and maximizing the domain classifier loss. Accordingly, the weights are updated during gradient descent as given below:

$$\theta_f = \theta_f - \mu \left(\frac{\partial L_y}{\partial \theta_f} - \lambda \frac{\partial L_d}{\partial \theta_f} \right) \tag{4.9}$$

$$\theta_y = \theta_y - \mu \frac{\partial L_y}{\partial \theta_y} \tag{4.10}$$

$$\theta_d = \theta_d - \mu \lambda \frac{\partial L_d}{\partial \theta_d} \tag{4.11}$$

where μ is the learning rate.

While the architecture explained here provides a generic prescription for the implementation of an arbitrary DANN architecture, the choice of hyperparameters such as number and types of layers, activations, and loss functions per component (feature extraction, label prediction, domain discrimination) depends on the task, experience, and expected performance.

4.3 Evaluation, Results, and Analysis

For the evaluation of the proposed domain adaptation approach for SHM, two case studies are analyzed. The first case study investigates the prediction performance for the damage condition of a gearbox system under various torques. In the second case study, a three-story structure with several levels of damage conditions is used.

4.3.1 Case Study 1: Gearbox Fault Detection

4.3.1.1 Dataset and Preprocessing

PHM Data Challenge 2009 introduced a dataset simulating various fault types for a generic gearbox system Phm Society (2009). Acceleration data were collected at the input and output shaft of the gearbox at different shaft speeds (30, 35, 40, 45, and 50 Hz) under two different loading conditions (low- and high-load). For each shaft speed and loading conditions, 6 fault types are simulated (normal, chipped gear tooth, broken gear tooth, bent shaft, imbalanced shaft, broken gear tooth with bent shaft). For each case which is the combination of fault type, shaft speed, and load condition, about 4 seconds of data is collected at a sampling rate of $f_s = 66.67$ kHz twice. *In this study, only the output shaft vibration data is considered.*

According to the literature on gearbox fault detection (Chen et al., 2015; Jing et al., 2017), the frequency domain provides a rich feature set for fault detection using vibration data. Thus, before training, all raw data is converted to the frequency domain using sliding-window Fast Fourier Transformation (FFT) also known as Short-Time Fourier Transform (STFT). The parameters for the transformations are selected as prescribed by the length of each window segment which is 1000 samples. The segments overlap by 80 percent and the sample length of FFT is 1200. The frequency resolution is $\Delta f = 111$ Hz. After prepossessing, each damage condition case has about 2700 data points with 601 features per loading condition. Here, each feature represents a STFT value corresponding to a frequency point discretized with Δf (i.e. $f_s/\Delta f = (66.67kHz)/(111Hz) \approx 601$). The dataset is divided into source and target domains according to loading conditions. The source domain corresponds to low loading conditions consisting of all shaft speeds and fault types whereas the target domain is composed of the high-load operation. Since the task is detecting the type of the fault regardless of shaft speed, the data belonging to the same fault type are stacked together. Finally, both domain data are split into training and test data using a 4-to-1 ratio. All data is standardized with respect to the source training data and all labels are one-hot encoded.

4.3.1.2 Implementation

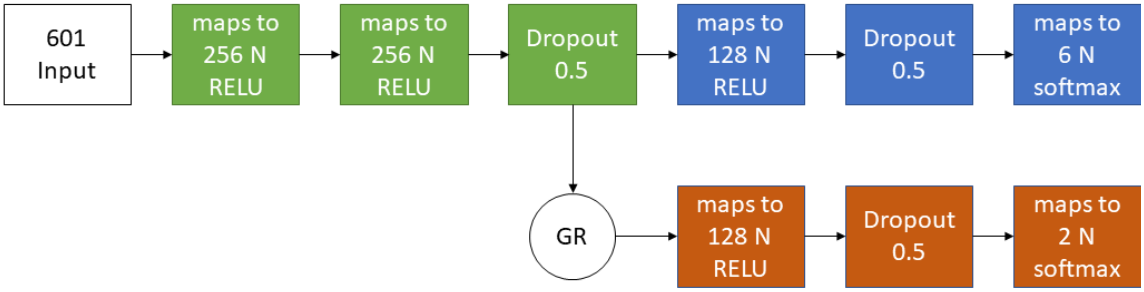
Three different models are developed: *Model 1:* source-only model which is trained only with source domain data; *Model 2:* the multi-tasking DANN model for training which uses both source and target domain data

to discriminate the domain and predict the label; and *Model 3:* single-task DANN model for prediction and used only for testing. The architectures are shown in Figure 4.3.

Here, each box corresponds to a layer of the neural network. *Input* is the input layer that utilizes the features. *N* is number of neurons. In other words, *256 N* within a box means 256 densely connected neurons are used for the particular layer. *RELU* stands for Rectified Linear Unit, and it is the activation function for the neurons defined within the layer. *Dropout* layers are represented along with the rate of dropout. For example, *Dropout 0.5* means a layer with a dropout rate of 0.5 is used. Finally, *Softmax* is the softmax activation layer that provides the class membership probability for each class.

The source-only model is a shallow network consisting of feature extraction (FE, colored in green) and class prediction (CP, colored in blue) layers. In addition to FE and CP layers, the multi-tasking DANN model includes the domain discriminator (DD, colored in red) layers and the gradient reversal (GR) layer. The single-task DANN model has the same structure as the source-only model but with updated weights where the FE contains the latent features that represent both source and target domains after training. Model 1 and Model 2 are trained using stochastic gradient descent. All the losses are chosen as categorical cross-entropy.

(a) Source-only model



(b) DANN model

Figure 4.3: Source-only and DANN architectures for numerical example

The low loading condition data represents the source domain whereas the high loading condition data corresponds to the target domain. During training, the DANN utilizes 128 data points (64 from source and 64 from target) per batch. We assume we have access to the source data labels but not to the target domain labels. The source (input, label) tuples are used explicitly for the class prediction task. For domain prediction, the source data is labeled as *0* and target data as *1*, and then the labels are one-hot encoded. The domain predictor uses both domain data for training and creating domain invariant features. The source-only model is trained with 75 epochs whereas the DANN is trained for 200 epochs.

In addition to DANN, TCA and JDA are used for comparison. TCA utilizes training data from both source and target domain to realize dimension reduction using radial basis function as the kernel. After dimension reduction, a support vector machine (SVM) classifier is trained on the labeled source data as prescribed by Pan et al. (2010). This classifier is also used to predict labels on unlabeled target data. Like TCA, dimensionality reduction procedure is applied on the training data from both domains for JDA. After feature transformation, a 1-Nearest Neighbor classifier (kNN classifier with $k = 1$) is trained on the labeled source data as prescribed in Long et al. (2013). This classifier is tested on the unlabeled target data used for training. Due to the way JDA is implemented, the algorithm cannot be tested on a new dataset other than the one it is trained with. Testing JDA on labeled source data will result to 100% accuracy. Results for source data indicate the perfect accuracy as N/A (not available).

Since TCA and JDA are essentially a set of matrix multiplications and eigenvalue decompositions, the complete training dataset does not fit into the memory. Due to this limitation, only a quarter of training samples are used from both domains. Both TCA and JDA methods are only applied to the first case study and then discarded for the second case due to their low performance.

4.3.1.3 Results

Table 4.1 shows the accuracy for source and target domain test data on the source-only model and the DANN model. The accuracy for predicting the source data is about 97 percent for both models. The accuracy for both source-only and DANN models on the training data is above 99% (not reported in the table). However, the generalization gap between training and testing is small which is an indicator for minimal overfitting.

Without domain adaptation, the accuracy of the target data for the source-only model is 64 percent. The DANN improves the prediction on target data and increases the accuracy to 71 percent. TCA method produces lower accuracy for source and target domain data ranging between 42 to 63 percent. Due to its limitation, JDA is tested only on training source and target data. JDA produces 100% accuracy for source data, and it is donated as N/A. JDA's domain adaptation performance for target data is around 42% and it is marginally lower than TCA.

The accuracy per class is visualized in Fig. 4.4. Accuracy plot shows that DANN performance improves significantly for some classes over source-only model. For class 3, the performance of both models is similar. A negative transfer learning is observed for class 4 and 5. As a result of this, the prediction performance of DANN is decreased. For all most all classes, TCA and JDA do not exhibit a good domain adaptation. Additionally, Table 4.2 illustrates the precision, recall, and F1 scores for all the models presented. Among all the models, DANN has the highest precision, recall, and F1 scores despite the slight negative transfer learning.

Model	Input	Accuracy
Source-only Model	Source	97.46%
	Target	64.43%
DANN Model	Source	97.55%
	Target	71.79%
TCA Model	Source	61.57%
	Target	45.25%
JDA Model	Source	N/A
	Target	42.08%

Table 4.1: Domain adaption performance for gearbox fault detection

Overall, results indicate that DANN outperforms TCA and JDA significantly and this finding implies that TCA and JDA may not be used as a reliable domain adaptation method.

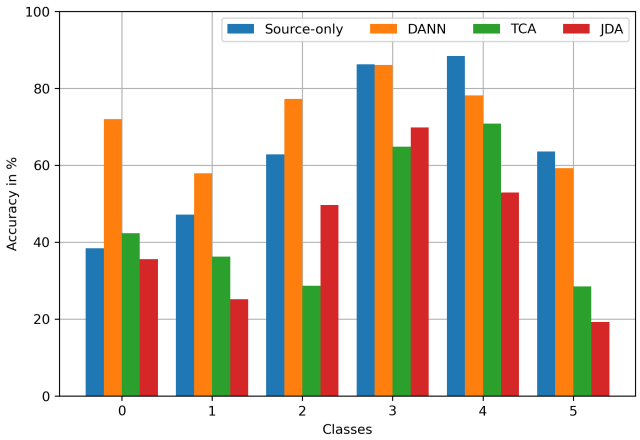


Figure 4.4: Accuracy per class

Model	Precision	Recall	F1
Source-only	65.81	64.47	64.59
DANN	72.06	71.80	71.90
TCA	45.24	45.25	44.10
JDA	41.82	42.09	41.06

Table 4.2: Additional performance scores for gearbox fault detection

In addition to the domain adaptation performance, we examined the empirical computational cost of each approach. Table 4.3 illustrates the average runtime for each model when they are trained 10 times. Among all, source-only model takes about 75 seconds to train over the entire dataset. The training time for DANN is longer (455 sec) as expected, since it utilizes a larger data set (source and target), and it is a multi-task

learning environment. Despite TCA and JDA use only a quarter of the data for training, it takes considerably a lot longer to train them. DANN employs stochastic gradient descent (SGD) for the optimization of the model weights. Thus, there is no need to load all the training samples in the memory at once. However, TCA and JDA require all the training data in the memory. In addition, the eigenvalue decomposition that TCA and JDA rely on is a set of matrix multiplications that are known to be computationally expensive for large dimensions.

Model	Average Training Time (sec)
Source-only	75
DANN	455
TCA	6470
JDA	1520

Table 4.3: Empirical computational costs for domain adaptation methods

4.3.2 Case Study 2: Structural Damage Detection

4.3.2.1 Structure and Numerical Model

This case studies the performance of domain adaptation when the training data are generated using a numerical model but the testing data are from an experimental structure. A small-scale three-story structure is tested by Figueiredo et al. (2009) at the Los Alamos National Laboratory (see Figure 4.5). The structure is excited with an electromagnetic shaker attached to its base. The accelerations at each floor including the base are recorded at a sampling rate of 320 Hz for about 25 seconds. Seven damage conditions are considered where the stiffness of one or two out of four columns at different stories are reduced. Table 4.4 summarizes the damage conditions.

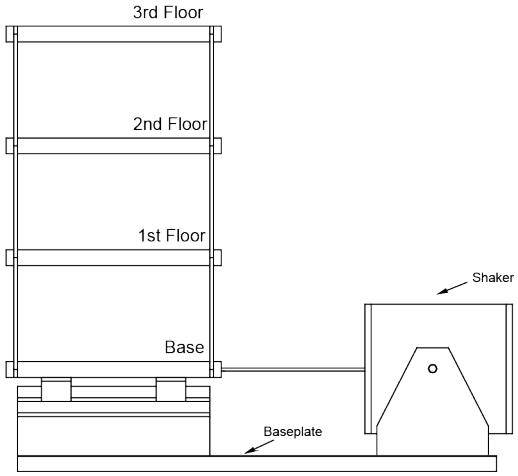


Figure 4.5: Three-story laboratory structure (Figueiredo et al., 2009)

Label	Damage Type
State #1	Baseline condition - Undamaged
State #2	87.5% stiff. red. in one column at first floor
State #3	87.5% stiff. red. in two columns at first floor
State #4	87.5% stiff. red. in one column at second floor
State #5	87.5% stiff. red. in two columns at second floor
State #6	87.5% stiff. red. in one column at third floor
State #7	87.5% stiff. red. in two columns at third floor

Table 4.4: Damage types for three-story structure

Our purpose for this case is developing a domain adaptation architecture that can transfer knowledge from the numerical model to experimental structure and predict the correct damage case. For this reason, we considered two sets of lumped-mass numerical models of the structure (see Table 4.6). First set is named as *low-fidelity model*, and it consists of two sub-model. The first sub-model, also called as *Simple*, establishes a 3-DOF system with the structural and geometric parameters provided by Figueiredo et al. (2009). A comparison of natural frequencies identified by Sun and Betti (2015) and extracted from *Simple* model is given in Table 4.5. The average error of the natural frequencies for the given model is about 33 percent, and the largest error is 50 percent occurring for the first mode. The differences in the parameters imply that the *Simple* model does not represent the actual structure.

The second low-fidelity model is named *Updated*. The model updating procedure prescribed by Giraldo et al. (2004) is performed on the *Simple* model to obtain this model. The method imposes a new stiffness matrix by utilizing the identified natural frequencies while the mass matrix remains the same. The *Updated* model has the same natural frequencies as the experimental structure, however, it is still 3-DOF. Thus, in this research, this model is still considered *low-fidelity*.

	Identified (Hz)	Simple (Hz)	Error (%)
1st Mode	31.09	14.32	53.94
2nd Mode	55.05	40.12	27.11
3rd Mode	72.23	57.98	19.73
		Average Error (%):	33.59

Table 4.5: A comparison of identified natural frequencies and Simple Model

The second model category is called *high-fidelity*. This model is 4-DOF and its design parameters such as mass and stiffness values are taken from Sun and Betti (2015) such that the natural frequencies and mode shapes of the model match closely to the experimental structure. To evaluate the sensitivity of modeling errors on the quality of damage prediction, two additional *high-fidelity* models are considered. The stiffness matrix of these models is perturbed by 5% and 10%, respectively, to simulate modeling errors.

All numerical models are simulated at 320 Hz in MATLAB. For these simulations, the base of the numer-

ical model is excited with acceleration that matches the dynamic characteristics of the data captured at the base level from the experimental structure. For each damage case, the numerical model is damaged by the values given in Table 4.4. The three floor acceleration responses from both numerical and experimental data are sliced into 1-second bins. The numerical and experimental data have a dimension of $[42350 \times 320 \times 3]$ and $[8750 \times 320 \times 3]$ (# of data instances $\times$ # of samples $\times$ # of channel), respectively. After the data is split into training and testing, it is standardized, and the associated labels are one-hot encoded.

4.3.2.2 Implementation

For this case study, a slightly modified version of convolutional neural network architecture proposed by the Lin et al. (2017) for structural damage detection is used as the source-only model. In addition to 1D convolutional and max-pooling layers, this network utilizes batch normalization for stable learning and dropout for regularization. For DANN implementation, the domain classifier is added just before the label predictor, which is a set of densely connected layers (see Figure 4.6). The output shape of each layer set is provided at the corner of those layers. The gradient reversal layer is not shown to keep the illustration simple.

For each numerical model, both architectures are trained from scratch. Source-only architecture is trained with the labeled numerical model data (source), and DANN utilizes both labeled model data (source) and unlabeled experimental data (target). For each case, the architectures are trained three times, and the average performance is computed. The source-only and DANN models are trained for 200 and 50 epochs, respectively. It is observed that more epochs do not improve the performance of the DANN model.

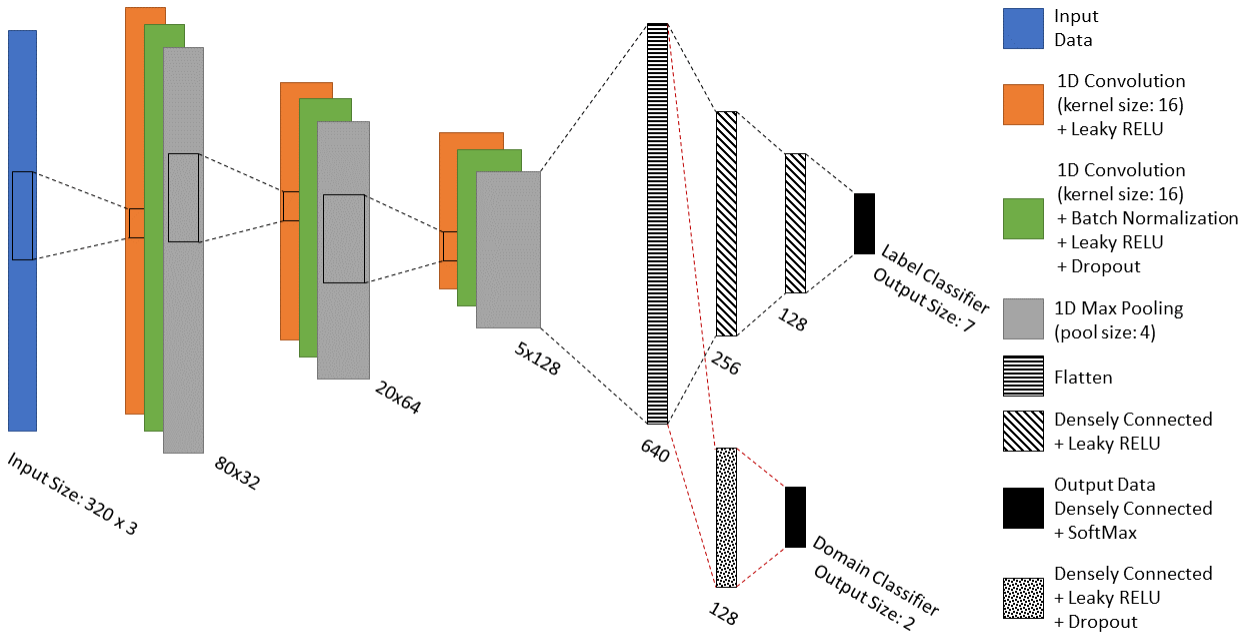


Figure 4.6: DANN architecture for experimental example

4.3.2.3 Results

Table 4.6 presents the damage classification performance of source-only and DANN architectures for the five numerical models. Additionally, for each model, the improvement in accuracy from source-only to DANN architecture is given as the difference between their target prediction performance.

First, we look at the performance of low-fidelity models. The accuracy of both architectures is high ($> 90\%$) for the Simple Model against source data. However, the accuracy drops to 14% when tested against target data.For Simple Model, the domain adaptation is ineffective when the modeling error is significant and the numerical model is semantically very different than the experimental structure. As for the Updated Model, the accuracy of source-only model marginally increases against target data (17%). The improvement is more prominent for DANN (40%) which is above 100% increase in the performance compared to source-only architecture.

For the high-fidelity model category, three models are considered with various modeling errors: 0%, 5%, and 10%. Both source-only and DANN architectures perform excellently against source data independent of the model ($> 98\%$ accuracy). The overall results for source-only architecture and no modeling error show that while the high-fidelity model is not perfect, it can predict the correct classes for the experimental data with an accuracy reaching up to 90%. When tested against target data, the classification performance of source-only architecture decreases progressively with respect to the modeling error. The largest performance degradation is observed (81% accuracy) when the modeling error is 10%. When there is no modeling error, domain adaptation improves the damage detection marginally compared to source-only model (88% vs 89%). For small modeling error (5%), the classification accuracy increases slightly (86% vs $\sim 90\%$). Lastly, the most dramatic increase in performance is observed for large modeling error where the improvement is higher than 10%.

		Low-Fidelity Model		High-Fidelity Model		
Architecture	Data	Simple	Updated	0% Error	5% Error	10% Error
Source-only	Source	96.33	94.01	98.99	99.22	99.21
	Target	14.40	17.08	88.35	86.49	80.88
DANN	Source	92.60	99.62	99.98	99.98	99.81
	Target	14.37	41.27	89.70	89.96	92.18
Improvement		-0.02	24.19	1.36	3.47	11.30

Table 4.6: Accuracy for domain adaption from numerical to experimental data (all values in percentage)

4.3.3 Discussion

To demonstrate the applicability of DANN, we consider two case studies. In the first case study, we predict the condition of a gearbox system running under high-load using the knowledge gained from low-load and high-load operation data. The second study focuses on transferring inference from labeled simulation data to unlabeled experimental data for a three-story structure. For the first case, the improvement DANN provides is around 7%. Here, TCA produced low accuracy both for source and target domain data. Similarly, JDA did not exhibit a successful domain adaptation performance either. This could be attributed to the fact that only a quarter of the total data set (generated with stratified random sampling) is used for the training. Similar to Principal Component Analysis, TCA and JDA are taxing on the memory for large number of samples. Due to this limitation, the generalization over both source and target data set may not be well defined. Additionally, TCA and JDA use SVM and k-NN on dimension-reduced source domain datasets, respectively. SVM and k-NN may not be the most suitable classifier for this application.

For the second case, we observe an increase in the target accuracy varying between 1% and 24%. The results show that if there is a significant divergence between source and target domains (Simple Model), domain adaptation is not very successful. For semantically similar systems (Updated Model), the learning model produced with the numerical data fails to predict correct labels for the target data without proper domain adaptation. On the other hand, the DANN is able to improve the accuracy of the target data by 24%. For high-fidelity models, the performance of source-only model decrease with increasing modeling error. DANN improves the target prediction performance compared to the source-only model, especially when the modeling error is more prominent. Due to the poor performance of TCA and JDA in Case 1, we dismissed them for Case 2.

4.4 Conclusion

For many SHM methods based on supervised learning, experimental target data is often not available. For such cases, a classification model trained with simulation data may not generate correct predictions for real data. Without addressing the data shift between the source and target domain, it is challenging to learn a model that can be used for SHM. This chapter shows that domain adaptation is a viable approach to damage classification problems. Specifically, we show the applicability of adversarial domain adaptation using two case studies. In the first case, we study the fault detection performance for a gearbox system between low-load (source) and high-load (target) domains and we observed that the prediction accuracy improves using domain adaptation. Additionally, we compared DANN to TCA and JDA to demonstrate the performance gain from DANN over the traditional domain adaptation methods. The second case focuses on detecting and locating damage for a three-story structure. Here, we utilized a numerical model of the structure for generating labeled

source domain data and the experimental data for unlabeled target domain data. The results show that DANN increases classification performance.

The current approach processes source and target data separately during training. In reality, for the majority of structural health monitoring applications, the structure is expected to be in healthy condition right after the construction. Thus, target domain data labeled as *normal/undamaged* is accessible for training to some extent. For future research, novel domain adaptation methods should exploit this limited target domain data during training to extract more generalized latent features and to improve the adaptation. Lastly, other domain adaptation strategies such as GAN-based discriminate approaches Tzeng et al. (2017) should be also explored.

CHAPTER 5

Model-based Damage Detection through Physics-guided Learning for Dynamic Systems

5.1 Introduction

In the last decade, the use of machine learning (ML) algorithms gained a lot of interest within the community of condition monitoring for dynamic systems (Widodo and Yang, 2007; Farrar and Worden, 2012; Stetco et al., 2019). A majority of ML applications in this area exercise a data-driven black-box approach that utilizes a large volume of experimental data obtained directly from the actual dynamic system. Black-box methods are proven to be successful in diagnosing the system through characterization and localization of the damage (Bakhary et al., 2007). One of the obstacles for such methods is often the availability of sufficient training data (Zhang and Sun, 2021). More specifically, access to a complete training dataset covering a wide range of conditions is costly and in some instances impossible without actually damaging the system prior to operation. This problem is a major roadblock in developing efficient data-driven algorithms for diagnostics of dynamic systems (Sadoughi and Hu, 2019).

For cases where training data captured from the field is limited, a data-driven black-box ML model could be trained with simulation data. In other words, to compensate for the lack of experimental training data, a representative analytical model can simulate the behavior of the system physics to some degree. While physics-based analytical models are capable of generating extensive training dataset, the resulting ML algorithm should still be evaluated with experimental testing data. Well-established analytical models are capable of simulating the dynamic response of the target system (Teughels and De Roeck, 2005; Jaishi and Ren, 2006). On the other hand, calibrating a large set of parameters for complex systems to achieve accurate physical behavior is often computationally exhaustive and at times infeasible (Zhang et al., 2020). Eventually, the analytical representation inherits modeling error. In this case, it is expected that the ML algorithm will fail to perform efficiently during testing since the simulation training data and experimental testing data are statistically divergent (Gardner et al., 2020). To address this drawback of data-driven black-box algorithms, the inference should incorporate domain-specific physical knowledge. The physics-guided learning (PGL) which is essentially a hybrid approach aggregating data-driven inference with physical parameters has the potential to leverage the performance of the condition monitoring further and to bridge the gap between simulation and experimental domains.

In recent years, a number of PGL approaches have been proposed (Karpatne et al., 2017; Jia et al., 2018; Sadoughi and Hu, 2019; Zhang et al., 2020; Zhang and Sun, 2021; Yao et al., 2020). However, the variety of

applications implies that the implementation of a proper PGL with domain-specific knowledge is non-trivial. Moreover, most of the existing work focuses on the prediction of system responses using PGL. In the area of diagnostics, little research effort has been devoted to incorporating of physical knowledge into the data-driven ML and exploiting deep learning architectures for damage classification.

In this chapter, we propose a damage detection and localization architecture for dynamic systems, namely, physics-guided learning for structural health monitoring (PGL4SHM) that combines the power of neural networks with domain specific physics knowledge. In particular, PGL4SHM is a multi-task deep learning architecture which (i) utilizes the synthetic data simulated by a numerical physics-based representation of the target structure for training and (ii) incorporates domain-specific physical parameters derived from this representation into the loss function. The multi-tasking PGL4SHM is trained with simulated structural responses in time-series form which serve as the input to the deep network. Additionally, during the learning phase, the physical parameters such as natural frequencies and mode shapes that are known to be structural damage indicators (Kim et al., 2003) are used for training the intermediate layers of PGL4SHM (Muralidhar et al., 2019). The modal features (natural frequencies and mode shapes) can be extracted directly from the numerical model. Since the organic relationship between the physics-based model representation, structural responses, and damage state is embedded into the PGL4SHM, the architecture is capable of generalizing damage detection compared to black-box approaches. As PGL4SHM uses modal features derived from numerical model, this embedding is physics-based rather empirical.

To validate the PGL4SHM architecture, two case studies are considered. The first study is purely analytical and investigates the performance and efficiency of the proposed approach under ideal conditions since every aspect of the system is simulated. The second case study considers the experimental setup of a small-scale three-story structure tested at Los Alamos National Lab. The results from both the analytical and experiential case studies show that the proposed architecture has better generalization capability in localizing the damage compared to black-box models in the presence of modeling errors. The performance gain is more evident when the numerical representation deviates from the actual structure further. Lastly, we evaluated the explainability of the results by analyzing the relationship between structural responses, damage state, and the integrated physical parameters.

Overall, PGL4SHM combines data-driven machine learning with knowledge of physics. As a result of this, PGL4SHM has the potential to improve damage detection and localization for SHM applications and promises more accurate decisions and prioritization for maintenance operations.

In summary, the major contributions of this study are:

- A physics-guided learning architecture, PGL4SHM is proposed to generalize damage detection and

location prediction for dynamic systems.

- The proposed architecture uses physics-constrained intermediate variable layers that rely on physical parameters known to be statistically important features for damage detection such as natural frequencies and mode shapes.
- For proof of concept, the proposed method is evaluated by comparing damage localization performance to black-box models for numerical and experimental cases.Results show that the new approach improves prediction accuracy in the presence of modeling error.
- PGL4SHM improves the explainability of the results since the intermediate layers expose valuable information that is highly relevant to the physics of the target structure.

5.2 Problem Formulation

Structural systems can experience damage throughout their life-cycle. It is essential to detect and locate the damage earl-on before it progresses to a bigger failure. In this context, damage localization is a supervised classification problem.

In this study, we consider black-box deep neural networks that label raw input data in the form of acceleration time-series measurements according to the damage condition the structure is experiencing. The implicit assumption for the black-box model is that the training data is available for every expected damage condition. In reality, the training data is only available for no damage condition. To obtain training data for other damage conditions, the structure should be deliberately tarnished which is not practical. We can create a physics-based model of the structure and generate simulated data for various damage scenarios of interest. Accordingly, the black-box model can be trained with the simulated data and tested with experimental data after deployment. However, this approach is often not feasible since the physics-based simulation often has intrinsic modeling error. Due to the deviation between simulated training and experimental testing data, the black-box model will become ineffective in labeling the input correctly.

The problem considered in this chapter is locating the damage accurately in the presence of modeling errors. To address this problem, three challenges should be resolved. First, we need to create a physics-based representation of the target system based on the available data. This model should be used to generate simulated data. Secondly, we should design a deep learning architecture that is trained with simulated data but can generalize well for experimental data compared to the black-box model. Thirdly, we need to establish the physical parameters most relevant to the damage condition of the structure to integrated into intermediate layers of the architecture during inference. Ultimately, successful implementation of PGL4SHM should

generalize the prediction well for the cases where field data is limited and physics-based simulation has some modeling error.

5.3 Physics-guided Learning

For a given set of structural response measurements in the time domain, we are interested in predicting the damage condition of the structure. Such a predictor can be trained with a supervised learning approach since for each input, x there is a label, y corresponding to the damage state. One way to learn the mapping from x to y is by training a black-box feed-forward neural network. By utilizing nonlinear activation functions within neurons, this network allows us to expose the complex relationship between the structural responses and the damage conditions. While the black-box networks are capable of learning the latent feature space, they can fail to generalize the predictions for unseen observations.

For many dynamic systems, the data labeled as *normal* is often available when they are deployed. However, access to data relevant to damage conditions is limited without harming the system. The absence of experimental data can be compensated by simulating damage on a finite element model of the system and obtaining new input/output pairs. However, due to the poor generalization of black-box models, the predictor will suffer from the presence of modeling errors and label the given inputs incorrectly. To address the limitation of the black-box models, this chapter presents the physics-guided learning for structural health monitoring (PGL4SHM) which integrates the physical knowledge regarding the dynamic characteristics of the target structure into the deep learning architecture.

5.3.1 Overview of PGL4SHM Architecture

Damage occurring in a load-carrying member changes the dynamic characteristics of the structure (Balageas et al., 2010). Fundamental dynamic characteristics of a system can be described in terms of its modal parameters such as natural frequencies, f, and mode shapes, ϕ. These parameters can be obtained from time series data using frequency domain analysis techniques (Brincker et al., 2000). Prior research has shown that supervised black-box algorithms utilizing modal parameters in the input layer can predict damage detection and localization with success (Wang et al., 1997; Hakim and Razak, 2014). A number of literature specifically focused on the use modal parameters such as natural frequencies and mode shapes to predict damages in a more refined manner (Kim et al., 2003; Wang and Li, 2012). On the other hand, majority of the aforementioned approaches depends on the existence of full-range experimental data. In this study, to overcome the limitations of black-box models, we propose PGL4SHM where the domain-specific knowledge is embedded into the deep learning architecture through intermediate layers inspired from Muralidhar et al. (2019).

Here, we assume that the input is structural response measurements in the time domain obtained from

a physics-based simulated model, the output is the damage condition associated with the input data. The intermediate layers utilize physical variables to improve supervised learning to enable a rich and generalized representation of the target system and to improve supervised learning. The physics-based model is developed as a representative finite element model (FEM) based on the available experimental data obtained from the undamaged structure. The simulated training data is generated using this FEM for various damage conditions of interest, including no damage case. The modal parameters f and ϕ can be extracted directly from the FEM. A simplified layout of the architecture for training is given in Figure 5.1. In this architecture, the input

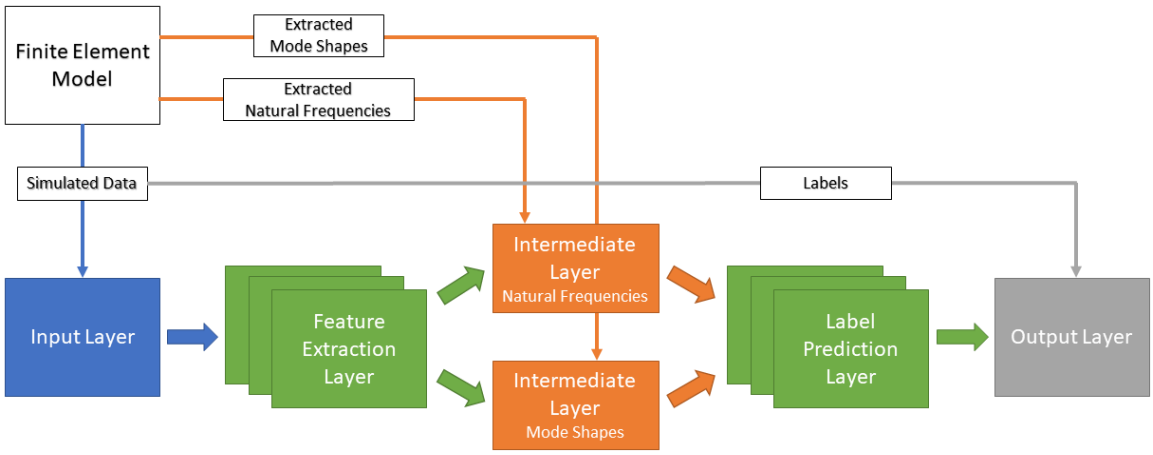


Figure 5.1: Simplified layout for training PGL4SHM

later takes the simulated time-series data obtained from FEM. Each piece of simulated data is associated with a label designating the damage condition. The feature extraction layers are a set of layers designed as convolutional neural networks (CNN). Additionally, there are two individual intermediate variable layers in parallel. The output of each intermediate layer corresponds to a modal parameter (f and ϕ). While the modal parameters are simply extracted from the FEM using eigenvalue problem (Craig Jr and Kurdila, 2006), it can be also derived from time series using domain-specific frequency-domain analysis processes (Ghanem and Shinozuka, 1995). The intermediate layers are *physics-guided* and are directly associated with physically meaningful modal parameters which are known to be good damage indicators. In this regard, this architecture exploits the feature extraction as a modal analysis step to compute intermediate variables which essentially blends domain-specific knowledge with the learning process. For this study, we assumed the intermediate variable layers are densely connected following a flattening layer after CNN based feature extraction layers. Next, the label prediction layers are tasked to extract features from modal properties to determine the damage condition for the given input. For training, PGL4SHM requires simulated time-series data, associated modal parameters, and labels. During testing, the architecture needs only experimental data to the input layer and predicts the relevant damage condition accordingly.

A black-box architecture is very similar to the PGL in nature with a main difference. Since black-box does not utilize intermediate layers and physical parameters associated with it, these layers are simply not implemented.

5.3.1.1 Physics-based Modeling

A physics-based presentation of the structure can be often achieved by modeling the target structure using a finite element model. In FEMs, the structural systems are modeled as a set of discrete elements (known as finite elements) that are related to the physical properties of the structure such as stiffness, area of the member section, etc. A finite element model can be idealized as a set of mass (M), stiffness (K), and damping matrix (C) which can be written in terms of equations-of-motions (EOMs).

$$M\ddot{x} + C\dot{x} + Kx = F \tag{5.1}$$

where F is the input excitation such as ground motion, x, $\dot{x}$, and x are the acceleration, velocity, and displacement obtained from the system, respectively. A finite element model can be generated either manually by representing each structural element faithful to its physical properties or by extracting the EOM matrices from the experimental data (Fritzen, 1986; Chen et al., 1996). Complex FEMs involve large dimensional matrices which complicate the modeling and make the calibration process cumbersome. As a result, the modeling errors are inevitable but often acceptable for many engineering applications.

Once the matrices are obtained, the training data can be simulated using Eq 5.1. By modifying M or K depending on the damage type, various damage conditions can be simulated. For every damage type, an eigenvalue problem can be applied to extract modal parameters as follows:

$$\lambda M\phi = K\phi \tag{5.2}$$

where λ is the diagonal eigenvalue matrix and can be written also as $\lambda = \text{diag}(2\pi f^2)$. It should be noted that for every combination of K and M pair, a unique pair of f and ϕ can be generated.

In addition to eigenvalue analysis, the modal parameters can be obtained from structural responses using sophisticated time and frequency analysis techniques (Ghanem and Shinozuka, 1995). A clear relationship between physics-based EOM matrices, structural responses, and modal features is obvious as all of them are related to the dynamic characteristics of the structure. Integration of modal parameters into the learning process as domain-specific knowledge is a promising tool for generalizing damage detection compared to black-box approaches.

5.3.1.2 Learning Process

This network is typically trained with structural response data obtained from a representative FEM. Additionally, the architecture utilizes physics-based modal parameters also obtained from FEM. Accordingly, the empirical loss function that needs to be minimized during learning can be formalized as follows:

$$Loss = Loss_{\text{DMG}} + \lambda_{\text{PGL}} Loss_{\text{PGL}} \tag{5.3}$$

Eq. 5.3 implies that the network utilizes a multi-task learning scheme, where $Loss_{\text{DMG}}$ corresponds to the categorical cross-entropy loss between the actual damage condition, y and predicted label, $\hat{y}$; $Loss_{\text{PGL}}$ represents mean square error (MSE) for the physics-guided learning parameters; and λ_{PGL} is the trade-off parameter. Please note that a black-box model will only use $Loss_{\text{DMG}}$ for the training and disregard the physics related loss, $Loss_{\text{PGL}}$.

The physics-guided loss, $Loss_{\text{PGL}}$ given in Eq. 5.3 can be described as:

$$Loss_{\text{PGL}} = Loss_{\text{PGL}}(f, \hat{f}) + Loss_{\text{PGL}}(\phi, \hat{\phi}) \tag{5.4}$$

Here, $Loss_{\text{PGL}}(f, \hat{f})$ is the MSE between the actual natural frequencies, f and predicted frequencies, $\hat{f}$; and $Loss_{\text{PGL}}(\phi, \hat{\phi})$ is the MSE between the actual mode shapes, ϕ and predicted ones, $\hat{\phi}$. Since $Loss_{\text{PGL}}$ is a regression loss, we assume the neurons of the intermediate layers are linearly activated.

5.4 Evaluation

For this study, we evaluated the PGL4SHM architecture by comparing it to the black-box model performance. We have considered two case studies. The first case study focuses on a finite element model of a simply supported beam, and the second case investigates experimental testing of a three-story structure.

5.4.1 Implementation

The FEM (simply supported beam) and experimental structure (three-story structure) are excited with white noise under various damage conditions and the resulting dynamic responses are collected from all available sensors in terms of accelerations for some amount of time. Then, the accelerations are divided into 1-second chunks and each of these chunks are categorized according to the relevant damage state. The data obtained from this process is the *reference data* and used for testing.

In parallel, for each case, another FEM model is developed to replicate the original structure. This model is intentionally misrepresented to some degree in order to introduce modeling errors that occasionally occur during the design process. Using this FEM, the structural responses and corresponding damage labels are

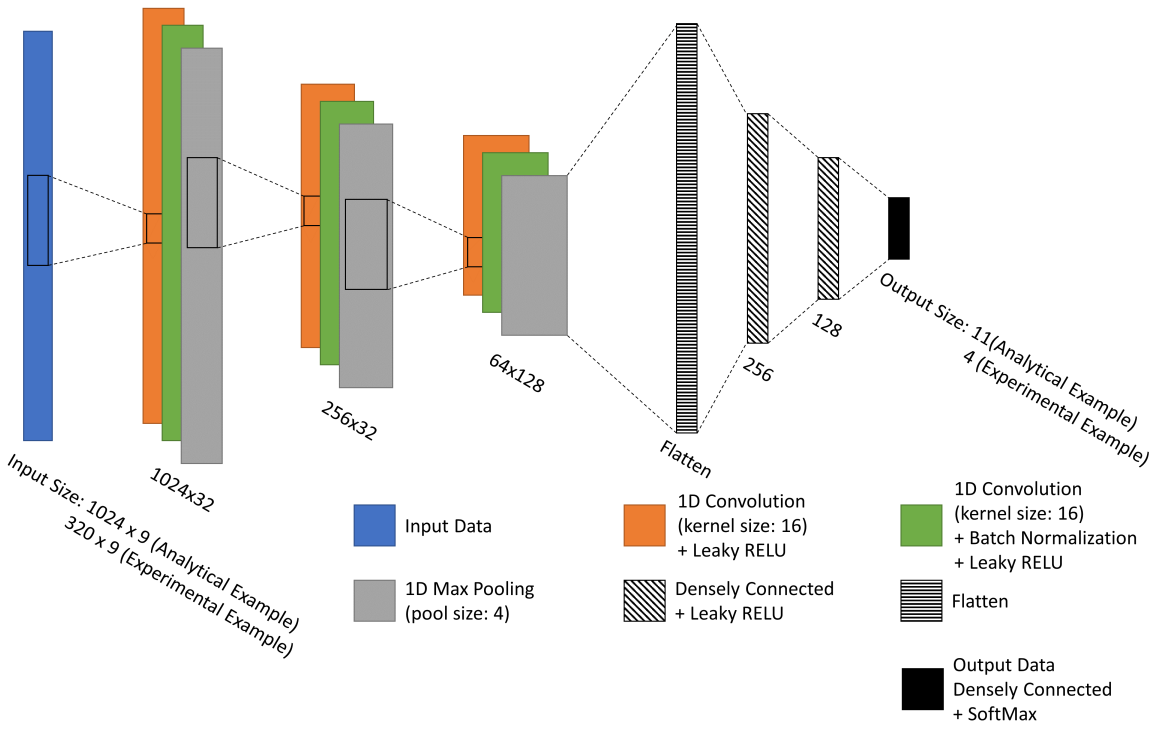


Figure 5.2: Black-box architecture adopted from Lin, Nie, & Ma, 2017

generated. In addition, the modal parameters are extracted from this FEM and vectorized. This data is then divided into training, validation, and testing with a ratio of $0.6 : 0.2 : 0.2$, respectively. The training and validation data is used during the training phase of PGL4SHM. The testing data and the *reference data* are used for performance evaluation. Before training, all available data is standardized by removing the mean and scaling to unit variance with respect to training data. All FEM and experimental data is standardized with scikit-learn toolbox.

Next, two neural network models are trained for each case. The first neural network is a black-box model that learns end-to-end relationship between the time series input and the damage condition (see Figure 5.2). The network is structured as prescribed in Lin et al. (2017). This model does not utilize physics-guded variables, mode shapes and frequencies at all. The dimension of the input depends on the number of the sensors and the sampling number. The feature extractor and label prediction layers are CNN and DNN, respectively. All neurons have leaky RELU activation functions. The size of the output layer changes with respect to the number of damage conditions considered for the case study. The neurons of this layer are activated with softmax. To generalize the predictions and mitigate the internal covariate shift, batch normalization layers are also inserted to the black-box model. Lastly, to reduce the number of trainable parameters, every batch normalization layer is followed by a max pool layer. The second model, PGL4SHM architecture is trained with the training data to minimize the loss function given in Eq. 5.3 (see Figure 5.3). This network struc-

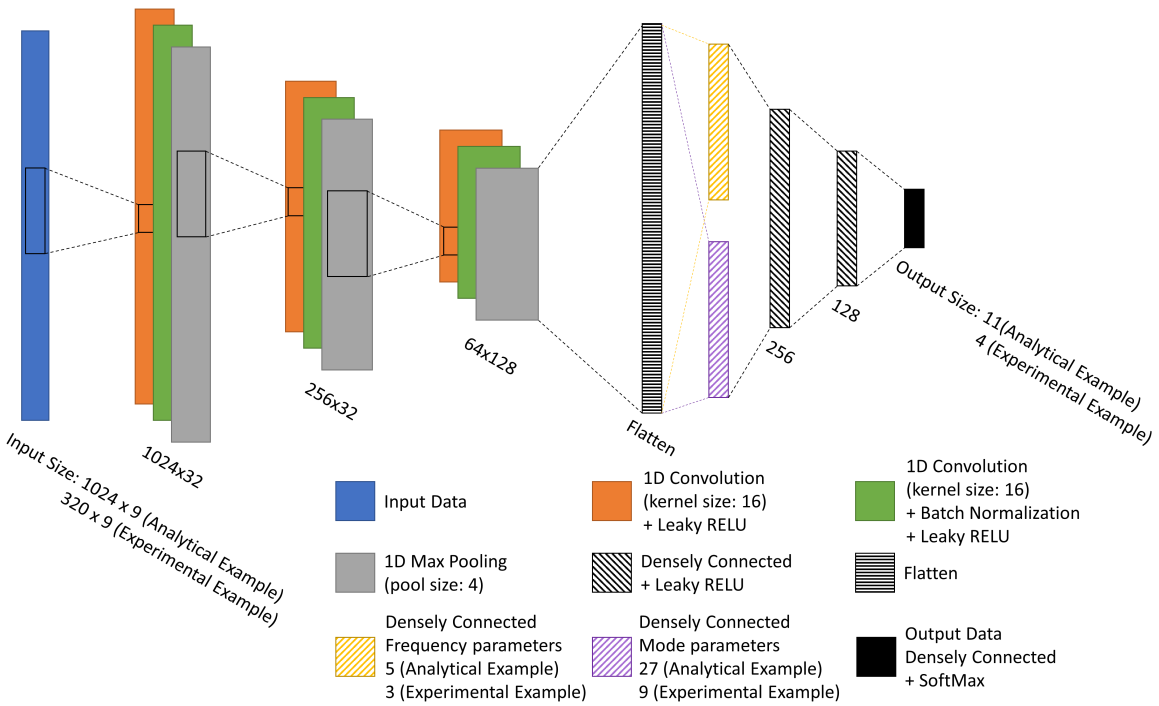


Figure 5.3: PGL4SHM architecture

ture follows an arrangement similar to the black-box model with the addition of intermediate value layers which employs the physics-guided modal parameters. The intermediate layers are densely connected and the neurons are linearly activated. Both black-box model and PGL4SHM architectures are trained using Keras running on TensorFlow 2.0 in Python 3.7. The performance of both architectures is evaluated by computing the classification accuracy.

5.4.2 Case 1: Analytical Example

This case focuses on the effectiveness of the proposed model where modeling errors relevant to environmental, operational, and material uncertainties are controlled more precisely. Here, we consider a simply supported beam studied by Lin et al. (2017). The beam has a span length of $L = 10.0\,m$ and a rectangular section with $0.1m$ width and $0.25m$ (see Figure 5.4). The beam is assumed to be made of steel with the elastic modulus of 206 GPa and density of $7,900\,kg/m^3$. The damping is simulated with classic Rayleigh damping where mass matrix (M) proportional factor, α is $1.0\,s^{-1}$ and stiffness matrix (K) proportional factor, β is $1.15 \times 10^{-6}\,s$. The beam is modeled using FEM tool, Open System for Earthquake Engineering Simulation - OpenSees (McKenna et al., 2010). The beam is discretized into ten equally long members that have linear elastic-beam column element properties. Excluding support nodes, the beam has 9 nodes. To generate acceleration responses, the beam is excited at each of the nine nodes vertically with a random noise.

This excitation has a Gaussian distribution with a mean of 200 N and standard deviation. To simulate finite features of the environmental noise, the random excitation is filtered with an eighth-order Butterworth filter that has a cutoff of 512 Hz. The sampling rate for the simulation is selected 8192 Hz. To reduce the volume of the data, the simulation data is down-sampled to 1024 Hz and only vertical accelerations at nine nodes are considered. For each loading case, the size of one simulation instance is $(9 \text{ nodes} \times 1024\ Hz)$. To simulate damage conditions, each of the ten members is damaged individually by reducing the member stiffness by 10% to 50% with 10% increments. Including *no damage state*, 11 damage conditions are simulated at nine loading positions across ten members. In addition to time-series data, for each data simulation, first 5 natural frequencies and 27 three modes shape points $(3 \text{ modes} \times 9 \text{ nodes})$ are extracted from the FEM analytically using OpenSees. The resulting data is categorized according to the damage location independent of the magnitude of the damage and the location of the excitation. All the data generated so far constitutes the reference data for testing. In parallel, another set of simulation data are generated with an inaccurate FEM model. To

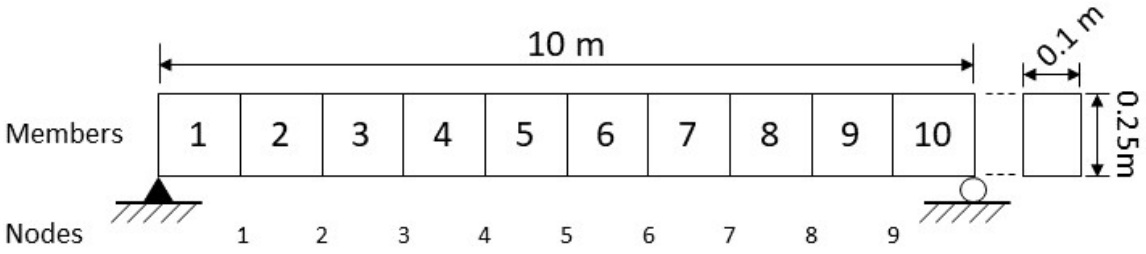


Figure 5.4: Simply supported beam model used for analytical case

account for environmental, operational, and material uncertainties, for each data instance, the stiffness of the inaccurate model is perturbed with a log-normal distribution. Four inaccurate models are developed where the maximum error of all sampled elastic modulus varies between 5% and 20% with 5% increments. This data is used for training, validation, and testing of PGL4SHM. In addition to time-series data, for each data simulation, the first 5 natural frequencies and 3 modes shapes are extracted from the FEM analytically using OpenSees.

Following the deep-learning architecture provided by Lin et al. (2017), the black-box model and PGL4SHM counterpart are developed, yielding about 1,072,267 and 621,739 trainable parameters to optimize, respectively. The PGL4SHM has intermediate layers between feature extraction and label prediction layers. These intermediate layers act as a choke point, decreasing the number of trainable parameters. To make up for the capacity of the PGL4SHM, two more convolutional layers (a regular convolutional layer and one with batch normalization and max pooling) are added before the flattening. This model, namely PGL4SHM - Extended, has 1,097,387 trainable parameters.

Two versions of PGL4SHM (regular and extended) are compared to the black-box architecture. Table 5.1

summarizes the classification accuracy and the improvement over black-box architecture with respect to the maximum modeling error in percentage. In addition, Figure 5.5 visualized the accuracy of all architectures. For no modeling error (ME 0%), while black-box outperforms the regular PGL4SHM, the performance of extended PGL4SHM surpasses all of them. When there is a small modeling error (ME 5%), black-box is the best among the three, resulting in to 94 percent accuracy. On the other hand, the difference between black-box and extended PGL4SHM (84.98 vs 84.55 percent) is negligible. The power of PGL4SHM shines when the modeling error is above 5 percent. For the cases ME 10%, ME 15%, and ME 20%, the performance of both PGL4SHM architectures succeeds black-box significantly. Overall, the improvement of prediction accuracy increases progressively with the modeling error.

In general, the black-box model is a good choice when the modeling fidelity is ensured. Both black-box and extended PGL4SHM have about the same amount of trainable parameters, and their prediction accuracies are similar. The extended PGL4SHM is successful for almost every case except the case ME 5%, however, compared to black-box, the performance loss is negligible. When the modeling error is small, compared to the extended PGL4SHM, the regular PGL4SHM is, in general, less effective due to the small number of trainable parameters. The results clearly show that especially when the numerical model does not represent the actual system properly, blending physical parameters with data-driven machine learning has a positive impact in improving the damage localization.

	Black-box	PGL4SHM		PGL4SHM - Extended	
Modeling Error	Accuracy (%)	Accuracy (%)	Improvement (%)	Accuracy (%)	Improvement (%)
ME 0%	93.75	91.90	-1.97	93.89	0.15
ME 5%	84.98	83.88	-1.29	84.55	-0.50
ME 10%	67.71	70.78	4.55	70.52	4.15
ME 15%	52.98	56.67	6.96	56.29	6.24
ME 20%	42.75	47.18	10.35	48.60	13.68

Table 5.1: Classification accuracy of black-box and PGL4SHM for analytical case

5.4.3 Case 2: Experimental Example

The performance of the PGL4SHM approach is also evaluated on a small-scale three-story structure tested by Figueiredo et al. (2009). An electromagnetic shaker is attached to the base of the structure (see Figure 5.6). The structure was excited with a band-limited white noise and the resulting horizontal acceleration responses and the excitation force were measured at a sampling rate of 320 Hz for about 25 s. For this study, including undamaged state, four damage conditions are considered. The damage states are established by reducing the stiffness of one or two columns at each floor by 87.5 percent. Each response data instance is categorized according to its respective damage condition. After the input force is removed from the measurements, time

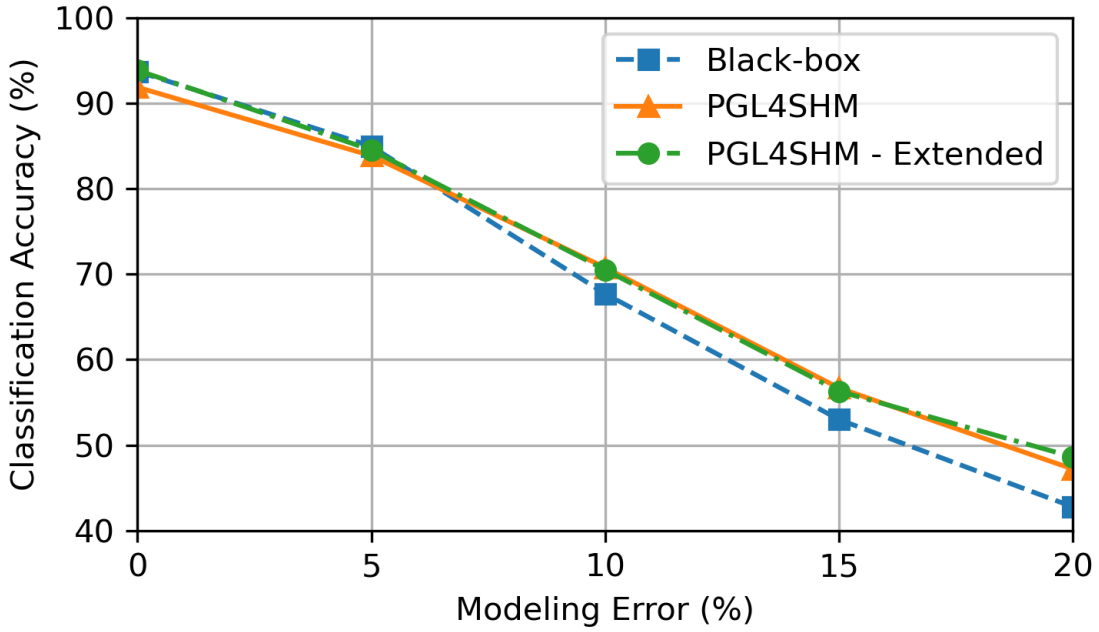


Figure 5.5: Visualization of classification accuracy for analytical case

series data are divided into 1-second chunks. Each chunk is categorized according to its respective damage condition. The data collected in this phase is the reference data for testing. In addition, a high-fidelity lumped-mass model is generated in the form of mass-stiffness-damping matrices using the parameters provided by Hernandez-Garcia et al. (2010) and Sun and Betti (2015). A 10 percent error is introduced into the stiffness matrix to simulate the modeling uncertainties. Using this imperfect model, data for all damage conditions are produced. In addition, three natural frequencies and 9 mode shapes points (3 modes $\times$ 3 stories) are obtained using this model. The data from the imperfect FEM model is used for training and validation of PGL4SHM. Black-box and PGL4SHM architectures have 707,844 and 557,459 trainable parameters, respectively. No further layers are added to PGL4SHM to extend the capacity of PGL4SHM. We also ensured there is no overfitting by validating the models against their respective numerical datasets. The performance of trained architectures and the improvement of PGL4SHM over black-box architecture for the corresponding modeling error are provided in Table 5.2. The classification performance of the black-box for no modeling error (96.18%) is greater than that of PGL4SHM (90.74%) and the performance loss reaches up to 6 percent. For a moderate level of modeling error (ME 10%), the black-box model yields a poor performance (38.06%) compared to PGL4SHM (70.82%).

Additionally, Table 5.2 presents the averaged F1-scores for the experimental case. The results and improvements are in parallel with the classification accuracies.

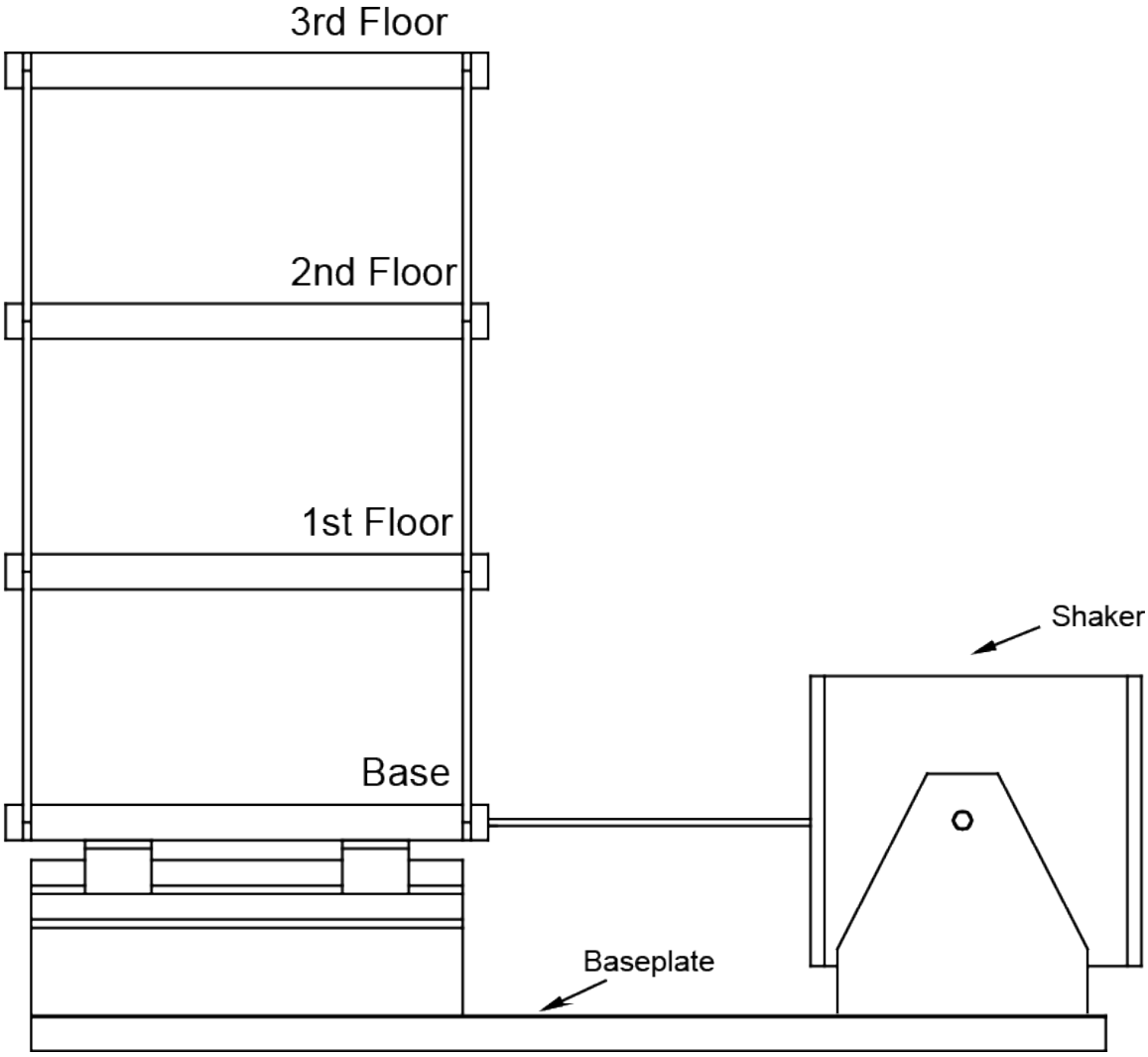


Figure 5.6: Three-story structure used for experimental case

	Black-box	PGL4SHM	
Modeling Error	Acc. (%)	Acc. (%)	Impr. (%)
ME 0%	96.18	90.74	-5.66
ME 10%	38.06	70.82	86.07

Table 5.2: Classification accuracy of black-box and PGL4SHM for experimental case

	Black-box	PGL4SHM	
Modeling Error	F1 (%)	F1 (%)	Impr. (%)
ME 0%	96.21	90.48	-5.96
ME 10%	31.09	67.36	116.66

Table 5.3: Averaged F1-score of black-box and PGL4SHM for experimental case

Lastly, Fig. 5.7 illustrates averaged ROC for the experimental case. Blackbox and PGL4SHM have similar ROC performance when there is no modeling error. Under moderate level of modeling error, PGL4SHM has a better classification performance compared to the blackbox model.

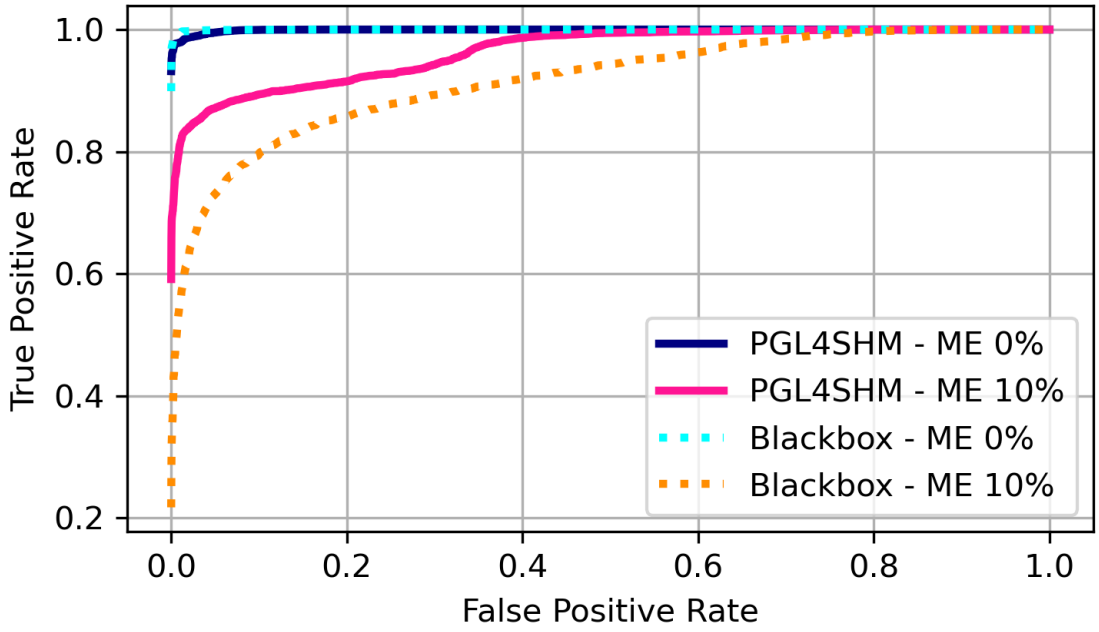


Figure 5.7: Averaged ROC curves for experimental case

From the results, it is evident that the black-box overfits the numerical data such that the latent features of the experimental data cannot be perceived. As a result, without the integration of the physical parameters, the data-driven black-box architecture fails to predict the damage classes correctly. The results indicate that in the presence of modeling error, the generalization of PGL4SHM is much more successful.

5.4.4 Effect of Hyper-parameters

Here, we investigated the effect of the trade-off parameter, λ_{PGL} on the prediction accuracy to understand the generalization of PGL4SHM under no modeling error. Table 5.4 summarizes the performance of both models and the weights for classification loss and physics-based loss for no modeling error. Trade-off parameter, in general, does not affect the performance of PGL4SHM, except for the case ($\lambda_{PGL} = 2.0$) where the weight for physical parameter loss is larger than the one for the classification loss. PGL4SHM with no weights to the physical parameters ($\lambda_{PGL} = 0.0$) is similar to the black-box model, but it still contains the intermediate layers. It is clear from the results that the introduction of intermediate layers degrades the performance of PGL4SHM when there is no modeling error. The small dimension of intermediate layers after the label prediction layer (see Figure 5.1) causes the learning to be *under-complete* leading to decrease in accuracy.

For larger models, the number of physical parameters can be increased and the label prediction layer will have a more complete basis for learning. For general purposes, weighting the losses equally ($\lambda_{PGL} = 1.0$) is a good starting point in training the PGL4SHM.

λ_{PGL}	Accuracy (%)
0.0	86.26
0.5	89.24
1.0	87.48
2.0	68.32

Table 5.4: Effect of hyper-parameters on the classification accuracy under no modeling error (ME 0%)

5.4.5 Interpretability of Intermediate Layer Outputs

We evaluated the explainability of the PGL4SHM by analyzing the relationship between the damage condition and intermediate layer outputs. Specifically, we focused on the interpretability of natural frequency, as it is more human-comprehensible and easier to visualize. Figure 5.8 illustrates *predicted* natural frequencies from intermediate layers, along with the *experimental* (true) and *simulated* (training) counterparts for four damage cases, where the modeling error is %10. Here, PGL4SHM is evaluated with experimental data. For each damage case, the intermediate layers in PGL4SHM predict three natural frequencies around 30,

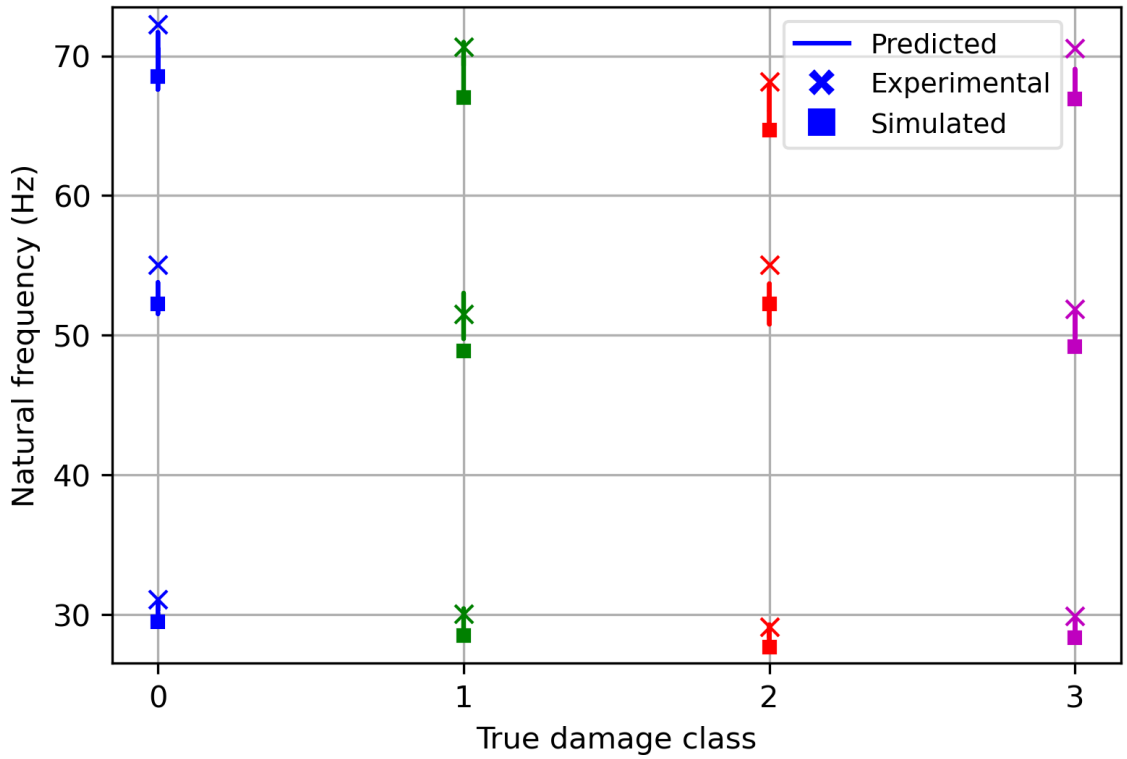


Figure 5.8: Interpretability of intermediate layer outputs

55, and 70 Hz with some variance. Compared to the *experimental* true frequencies of the structure (square markers), *simulated* values extracted from FEM (cross markers) always undershoot. This is expected since the modeling error is introduced to the FEM by reducing the stiffness matrix by 10 percent which causes the *simulated* frequencies to decrease. In general, the *predicted* frequencies range from *simulated* to *experimental* values.

During training, the simulated modal parameters are used for physics-based loss function. On the other hand, integration of physics-based parameters into the training also constrains the inference such that PGL4SHM favors to predict the modal parameters towards the experimental true counterparts. There are some cases where the predicted values do not distribute uniformly between experimental and simulated values. The distortion is substantial especially for the second modes (50 Hz) of damage class 1 and 2. This error causes some of the intermediate value outputs from class 1 and 2 to overlap with the damage class 0 (no damage class) leading to mislabeling. Due to the explainability of results, such problematic instances can be in theory captured algorithmically and corrected at testing time.

5.5 Conclusion

In this chapter, we have presented a physics-based deep learning architecture, PGL4SHM to detect and localize the damage in mechanical systems. The proposed approach incorporates physical parameters such as natural frequencies and mode shapes, which are known to be statistically meaningful features for damage detection, into the intermediate layers of deep neural networks. To accommodate the intermediate layers, the architecture introduced physics-based loss into empirical loss function. To evaluate the proposed approach, we considered analytical and experimental cases. Both examples show that physics-guided learning improves the accuracy of the damage localization compared to black-box models in the presence of modeling errors. Our empirical study shows that weighting the classification and physical loss equally is an effective starting point for training. Lastly, we discussed the interpretability of intermediate layer output by analyzing the relationship between predicted modal parameters and classification performance. Our findings indicate that the misclassified instances could be explained through the characterization of predicted natural frequencies.

CHAPTER 6

Physics-guided Deep Learning with Domain Adaptation for Structural Health Monitoring

6.1 Introduction

Recent advances in machine learning (ML) have transformed significantly the facade for research on structural health monitoring (SHM) (Farrar and Worden, 2012). The majority of contemporary SHM applications employ a supervised black-box ML to infer upon large volume of experimental data. While black-box methods are quite successful in diagnosing the structural health through characterization and localization of the damage (Bakhary et al., 2007), they are often limited by the availability of sufficient training data (Zhang and Sun, 2021). Availability of data is a major obstacle for the development of effective ML algorithms for diagnostic SHM applications and impedes the advancement and transformation of SHM (Sadoughi and Hu, 2019).

A significant portion of ML-based SHM applications focus on damage localization and formulate this task as a supervised classification problem (Avci et al., 2021). The majority of supervised damage localization approaches require substantial data for each class for proper training (Yu et al., 2019; Avci et al., 2018; Lin et al., 2017). In reality, a complete set of experimental data is often inaccessible outside the laboratory, which renders supervised methods impractical for real world applications. To address the lack of experimental data for training, research efforts have been directed to explore model-based damage localization methods (Kopsaftopoulos and Fassois, 2013). These approaches usually aim to develop a high-fidelity numerical model of the structure based on the data captured on the field and to generate a complete set of simulated data for training the ML (Moughty and Casas, 2017). However, issues commonly occurring in practice such the impact of implicit modeling errors due to model idealization on the generalization and the upkeep of the ML algorithm to address the changes in the structure are usually not acknowledged in the current literature (Gardner et al., 2021). It is clear that the research for state-of-the-art model-based supervised SHM applications should take a step towards improving robustness and generalization of damage localization tasks in such events.

A fundamental assumption for supervised learning is that the joint distribution of input-output is independent and identically distributed (i.i.d.) for training and testing data. In contrast, the phenomena of *changing environment*s is a known problem within the ML community due to its frequent appearance and it violates the i.i.d. assumption (Alaiz-Rodríguez and Japkowicz, 2008; Moreno-Torres et al., 2012). More specifically, the so-called *dataset shift* implies that machine learning algorithms are susceptible to the statistical shifts in the probability distributions between training and testing datasets leading to poor prediction

performance (Quiñonero-Candela et al., 2009). In the last decade, transfer learning has emerged as a new research topic to remedy the generalization problems due to dataset shift. In essence, transfer learning targets to improve the learning and inference by transferring the knowledge obtained from a labeled dataset, task, and model, namely, *source domain* to a new *target* domain with unlabeled dataset (Pan and Yang, 2009). Among many transfer learning approaches, domain adaptation has gained a particular interest within the SHM community as an effective method. Mainly, domain adaptation aims to capture a latent feature space that generalizes well over both source and target domains (Patel et al., 2015; Wang and Deng, 2018). As stated previously, a majority of ML-based SHM approaches rely on training labeled data generated using simulations and but need to be tested on realistic unlabeled experimental data. Statistical difference between source and target dataset, i.e. dataset shift due to modeling errors or unaccounted changes in the structure can harm the performance of such ML applications. Under dataset shift, domain adaptation presents itself as a viable tool for ML-based SHM to improve the generalization over both domains (Gardner et al., 2020; Ozdagli and Koutsoukos, 2020, 2021a; Lin et al., 2022). A recent addition to the literature in the area of damage detection by Lin et al. (2022) introduced the minimization of the maximum mean discrepancy (MMD) between source and target datasets as a loss function.

In the last few years, physics-guided machine learning started to exploit deep learning architectures as it provides a versatile platform for integrating the underlying physics knowledge specific to the problem into the learning process and improving the generalization capability under modeling uncertainty Karpatne et al. (2017); Jia et al. (2019); Sadoughi and Hu (2019); Yao et al. (2020). In general, physics-guided learning (PGL) incorporates the knowledge in the form of a physics-guided loss function. PGL is especially useful for model-based damage identification problems as the model used for training the architecture often has inherent modeling errors due to modeling idealization and simplifications Zhang and Sun (2021).

Some other early representative publications also look at the damage detection problem through model or physics-based parameter update (Sarma and Adeli, 2001; Ni et al., 2008; Amezquita-Sanchez et al., 2017; Oh et al., 2017; Perez-Ramirez et al., 2019; Pereira et al., 2020). However, the majority of these works assume that the algorithm developer has access to the labeled data to some degree. As mentioned previously, this study assumes that the availability of labeled data is often limited, i.e., the baseline for a data-driven numerical model is non-existent.

This chapter proposes a deep learning framework that combines the power of domain adaptation with the physics knowledge to constrain the learning process. More specifically, this architecture utilizes the so-called intermediate variable layers which store problem-specific physics-related latent information as well as the adversarial domain adaptation to learn a physics-informed domain-agnostic feature space representing both source and target domains. To achieve generalization over both domains, the proposed framework employs a

novel physics-based domain adversarial multi-task learning objective.

To validate the framework, this study considers a classification problem within the area of structural health monitoring for locating the damage on a small-scale three-story structure using its vibration response data. For this problem, it is assumed that the access to the labeled experimental data is limited. Therefore, a simulation model is developed to generate labeled training dataset which serves as the source domain data. Further, the framework utilizes modal parameters extracted from this simulation model which are known to be important structural damage indicators for training the intermediate layers. As a result, the relation between the simulation model, structural responses, modal properties, and damage location is captured into the intermediate variable layers. Combined with adversarial domain adaptation, the physics-constrained intermediate layers improve generalization of damage localization since it enables a physics-informed domain-agnostic feature space. The performance gain for the proposed framework over it alternatives is more evident in the presence of modeling errors.

While the proposed framework mainly focuses on one type of SHM problem, the architecture discussed in this chapter can be customized to explore other SHM applications where dataset shift is observed between source and target domains and learning can be constrained through the ingestion of physics-based knowledge.

The major contributions of this study are summarized as below:

- A new deep learning framework that combines the physics-guided learning with domain-adversarial training is proposed to leverage generalization.
- The proposed framework exploits physics-constrained intermediate variable layers based on parameters and features specific to the problem. In addition, adversarial training using domain adaptation ensures the formation of latent representations of both the source and target domains.
- Integration of intermediate layers and domain adaptation into the training procedure enables a physics-informed domain-agnostic latent feature space that generalizes well over both domains.
- To evaluate the proposed framework, an SHM problem focusing on damage localization is considered. Results demonstrate that the proposed approach improves the generalizability significantly even in the presence of modeling error.

6.2 Background

6.2.1 Motivating Example

Success of an informed maintenance and repair decision on a structure of interest depends on tracking the up-to-date health state of the structure. In typical structural health monitoring applications, the change of

a feature in structural responses such as the magnitude in time domain or the power density in frequency domain is accepted as a good indicator for detecting structural damages. To capture such anomalies, one can deploy sensors, i.e. accelerometers and measure vibration responses of the structure under its operational loads and compare this data with our prior knowledge.

In recent years, SHM community started to invest significant amount of research for improving damage detection method by utilizing modern deep learning tools (Sohn et al., 2002; Farrar and Worden, 2012). While damage detection in structures is a well-established problem that can be solved with traditional ML approaches (Gres et al., 2017), damage localization and quantification is still a critical problem.

In this chapter, the damage localization is posed as a classification task where the input is a set of structural response measurements collected from various sensors on the structure in the time domain and the output is the location of the damaged member. One approach to reveal a mapping function from input to output is training a black-box neural network. While the black-box networks can learn the latent feature space for the training dataset, they may have difficulty in generalizing over previously unseen test data. The overarching problem for the decline in performance can be attributed to the fact that the training dataset from the source domain D_S and the test dataset from the target domain D_T may be statistically different such that their probabilities are not equal, i.e. $P(D_S) \neq P(D_T)$. This probabilistic divergence between the domains can be explained through the following two phenomenon:

i. Lack of available experimental data:

Suppose that we want to train a supervised ML-based damage localization algorithm, G_y that takes an input sample x_i as the time series data and predicts a label for y_i which is the location of the damaged member. When a structure is deployed for service, it is assumed to be intact and in healthy condition. Therefore, the experimental data necessary for training the algorithm often lacks information about the damaged state of the structure. In addition, it is also not feasible to damage the structure at various locations just to generate the necessary dataset.

To counterbalance the lack of data, the algorithm designer can develop a numerical model of the target structure and simulate the responses for the damaged states of interest cost-effectively. After an ML algorithm is trained using this simulation data, it should still be evaluated with experimental testing data to ensure generalization. A main challenge for achieving accurate physical behavior is often hindered by the calibration of complex simulation models which can be time-consuming and exhaustive process due to modeling uncertainties (Zhang et al., 2020). As a result of this, the simulation data may have some modeling errors and may not be able to represent the actual physical behavior in high-fidelity. The statistical divergence between the simulation data (source domain) used for training the ML and the experimental test data (target domain) for

testing leads to a decrease in performance for predicting the correct damage location (Gardner et al., 2020)

ii. Shift in the dataset:

Structures are often repaired or rehabilitated during their life-cycle, even when they are not damaged. On the other hand, a change in non-structural elements such as resurfacing the road of a bridge may still alter the behavior of the structure (Devin and Fanning, 2012). Even if the algorithm designer has the capacity to generate training datasets from high-fidelity simulation models that represent the structure with minimal modeling error, due to the aforementioned fact that structures experience changes in time, there will be a shift in the dataset between source and target domains (Quiñonero-Candela et al., 2009).

6.2.2 Problem Definition

This chapter presents a deep learning framework that combines physics-guided learning with domain adaptation for structural health monitoring applications. In particular, this study considers a damage localization problem where a civil structure in service may experience some damage during its life-time. Here, the localization problem is posed as a classification task. In this task, X represents the input space which constitutes the time-series data collected from various sensors positioned on the structure. Similarly, Y denotes the output space which consists of possible labels corresponding to the damage location. It is assumed that the given labeled training dataset originates from source domain D_S, whereas the unlabeled test dataset is derived from target domain D_T. The source and target domains is defined as follows:

$$D_S \sim X_S = \{x_i, y_i\}_{i=0}^{n_s} \tag{6.1}$$

$$D_T \sim X_T = \{x_j\}_{j=0}^{n_t} \tag{6.2}$$

where n_s are n_t are the number of samples drawn from source and target domains, respectively. Our goal is to train a classifier on the source domain data that can generalize well for the target domain.

6.3 Proposed Deep Learning Architecture for Damage Localization

Before tackling the aforementioned problem, this section first introduces a general background knowledge on domain adaptation and physics-guided learning. Then, it discusses the proposed architecture that integrates both notions into deep learning.

6.3.1 Adversarial Domain Adaptation

One way to train a classifier that will generalize well on both domains is domain adaptation approach. For typical domain adaptation methods, our aim is to find a latent representation to minimize the probabilistic

divergence among the source and target. Domain adversarial neural network (DANN) attempts to reduce this statistical distance between two domains by augmenting a discriminator to a traditional deep learning based classifier (Ganin et al., 2016). Figure 6.1 illustrates a typical DANN implementation. Here, G_f represents the set of neural network layers that extract latent features. The label predictor G_y generates class label using the extracted features. Lastly, the discriminator G_d acts as a domain classifier and predicts the origin of domain for a given input. The parameters of feature extractor, label predictor, and domain classifier are denoted as θ_f, θ_y, and θ_d, respectively.

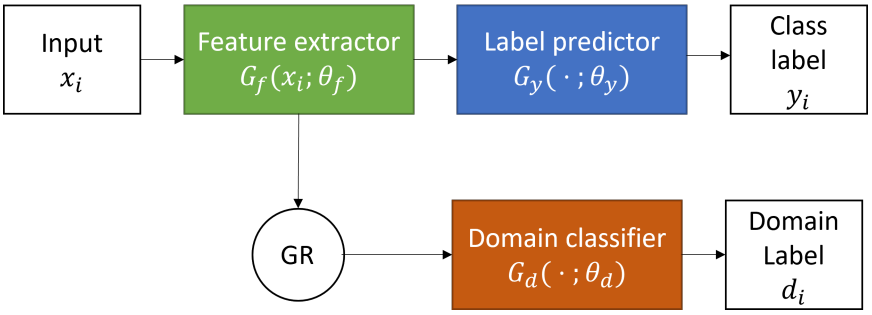


Figure 6.1: Concept of Domain Adaptation

The goal of DANN is to determine a domain-invariant feature space capable of confusing the discriminator such that it will not be able to tell the origin of domain. Accordingly, the training strategy of DANN has two tasks: (i) minimizing the class label loss $\mathscr{L}_y$; and (ii) maximizing the domain classifier loss $\mathscr{L}_d$. Then, this multitask learning loss to minimize can be formulated as following:

$$\mathscr{L} = \frac{1}{n_s} \sum_{x_i \in D_s} \mathscr{L}_y(y_i, \hat{y}_i) - \frac{\lambda_d}{n_s + n_t} \sum_{x_i \in D_s \cup D_t} \mathscr{L}_d(d_i, \hat{d}_i) \tag{6.3}$$

where λ_d is the trade-off parameter between the class label and the domain label loss, x_i is the i_{th} input; y_i and $\hat{y}_i$ are the true and predicted class labels, and d_i and $\hat{d}_i$ are the true and predicted domain labels.

In order to determine the weights θ_f, θ_y, and θ_d that minimizes the given loss, a forward- and backward-propagation procedure is implemented. During forward-propagation, the class label loss $\mathscr{L}_y$ is calculated using the dataset from source domain, while the domain classification loss $\mathscr{L}_d$ utilizes both the source and target domain datasets. During the backward-propagation, the gradients are computed first. Then, a pseudo function called Gradient Reversal (GR) layer (shown in Figure 6.1) is used to reverse the gradient by multiplying it with a small factor. In other words, the GR layer has distinct and different behavior for forward and

backward propagation (see Eq. 6.4 and Eq. 6.5) as given below:

$$GR(x) = x \tag{6.4}$$

$$\frac{dGR}{dx} = -\mathbf{I} \tag{6.5}$$

Eq. 6.4 implies that during the forward propagation, the GR layer acts as a throughput where the output of the feature extractor layer is the input to the domain classifier. During the backpropagation, GR acts as a co-factor in its partial derivative form as prescribed in Eq. 6.5 and multiples the gradient from coming from the domain classifier and passes to the next layer. Please note that GR affects only the gradients originating from the domain classifier and it does not alter the gradients from label prediction. The bi-modal behavior of GR function during forward and backward propagation ensures the goals of DANN, i.e. minimization of class label loss and maximization of the domain classifier loss.

Once the gradients are computed, the weights are updated using gradient descent as prescribed below:

$$\theta_f = \theta_f - \mu \left(\frac{\partial \mathscr{L}_y}{\partial \theta_f} - \lambda_{\text{DA}} \frac{\partial \mathscr{L}_d}{\partial \theta_f} \right) \tag{6.6}$$

$$\theta_y = \theta_y - \mu \frac{\partial \mathscr{L}_y}{\partial \theta_y} \tag{6.7}$$

$$\theta_d = \theta_d - \mu \lambda_{\text{DA}} \frac{\partial \mathscr{L}_d}{\partial \theta_d} \tag{6.8}$$

where μ is the learning rate. Here, θ_f is the weights for latent feature space that can generalize well over both source and target domain.

6.3.2 Physics-guided Learning

As an alternative to domain adaptation method, physics-guided learning has gained a lot of interest in treating dataset shift-like problems occurring in SHM. In essence, a typical physics-guided framework constrains the learning process by ingesting the physics-based knowledge into the neural network. In this chapter, the ingestion of knowledge is presented in terms of physics-based loss in addition to the classification loss as given below:

$$\mathscr{L} = \frac{1}{n_s} \sum_{x_i \in D_s} \mathscr{L}_y(y_i, \hat{y}_i) + \frac{\lambda_{pgl}}{n_s} \sum_{x_i \in D_s} \mathscr{L}_{\text{pgl}}(z_i, \hat{z}_i) \tag{6.9}$$

where $\mathscr{L}_y$ is the classification loss; $\mathscr{L}_{pgl}$ is the physics-based loss, λ_{pgl} captures the trade-off between the class label and physics-based loss; z_i and $\hat{z}_i$ are the physics-related true and predicted parameters. Accordingly, similar to adversarial domain adaptation, the physics-guided learning is also a multi-tasking learning process.

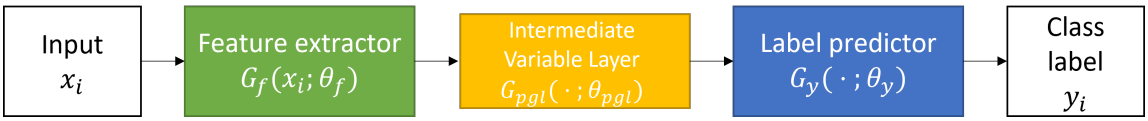


Figure 6.2: Concept of Physics-guided Learning

This chapter implements the physics-based loss by introducing intermediate physics-informed parameters (see Figure 6.2). More specifically, we select domain specific variables that can be obtained during data collection directly or indirectly and associate them with neurons within a layer. The intermediate layers use physical variables to allow a generalized representation (Muralidhar et al., 2019). In order to obtain the loss $\mathscr{L}_{pgl}$, the intermediate layers are tasked with making predictions upon the physical variables such that:

$$\mathscr{L}^i_{pgl}(\theta_f,\theta_{pgl}) = \mathscr{L}_{pgl}(z_i,\hat{z}_i) \tag{6.10}$$

$$= \mathscr{L}_{pgl}(G_{pgl}(G_f(x_i;\theta_f);\theta_{pgl}),\hat{z}_i) \tag{6.11}$$

Here, natural frequencies and mode shapes are chosen as the intermediate variables since they are known physical features that can be related to the location and the quantity of the damage observed (Yuen, 1985; Pandey et al., 1991; Kim et al., 2003; Ozdagli and Koutsoukos, 2019). These features can be extracted from a simulation model or eigenvalue analysis (Juang and Pappa, 1985). The physics-guided loss can be described as:

$$\mathscr{L}_{pgl} = \lambda_{\text{pgl},1}\mathscr{L}_{\text{pgl},1}(f_i,\hat{f}_i) + \lambda_{\text{pgl},2}\mathscr{L}_{\text{pgl},2}(\phi_i,\hat{\phi}_i) \tag{6.12}$$

where $\mathscr{L}_{\text{pgl},1}(f_i,\hat{f}_i)$ is the MSE between the set of true natural frequencies, f and predicted ones, $\hat{f}$; $\mathscr{L}_{\text{pgl},2}$ is the MSE between the set of true mode shapes ϕ and the predicted ones $\hat{\phi}$; and $\lambda_{\text{pgl},1}$ and $\lambda_{\text{pgl},2}$ are the trade-off parameters for the physics-guided losses such that $\lambda_{\text{pgl},1} + \lambda_{\text{pgl},2} = 1$. Since $\mathscr{L}_{\text{PGL}}$ is a regression loss, the intermediate layers use neurons with linearly activations. It should be noted that for the given PGL setting, the input to the label predictor is the output of the intermediate layers that form the input to the label predictor. The label predictor layers do not utilize the output values from feature extractor layers directly. Accordingly, the loss on the class label can be described as:

$$\mathscr{L}_y(y_i,\hat{y}_i) = \mathscr{L}_{pgl}(G_y(G_{pgl}(G_f(x_i;\theta_f);\theta_{pgl});\theta_y),\hat{y}_i) \tag{6.13}$$

A powerful and unique advantage of intermediate layers is improved interpretability. During testing time, the intermediate layers are capable of generating new information in the form of physics-related parameters ($\hat{z}$) which were previously inaccessible by the black-box model for unlabeled data. Since the intermediate layers

serve as a physics-based knowledge container, they can be used to extract new explanations to describe the input data. For this chapter, interpretability and explainability is out of scope and are not discussed further.

6.3.3 Integration of Domain Adaptation into Physics-guided Learning

In this chapter, we hypothesize that combining physics-guided learning with domain adaptation generates a physics-informed feature space embedded in the weight θ_f that can generalize well on both source and target domains. In particular, this study focuses on damage localization problems for SHM applications under model uncertainty or dataset shift caused by changes in the structure during its lifetime.

To support this hypothesis, this chapter proposes a deep learning framework that integrates domain adaptation into physics-guided learning. A conceptual illustration for this framework (or DAPGL for short) is provided in Figure 6.3. This architecture uses intermediate physics-related parameters as well as the adversarial domain adaptation to learn domain-agnostic feature space. The loss function for this framework can be formalized as following:

$$\begin{aligned}\mathscr{L} = &\frac{1}{n_s}\sum_{x_i\in D_s}\mathscr{L}_y(y_i,\hat{y}_i)\\ &-\frac{\lambda_d}{n_s+n_t}\sum_{x_i\in D_s\cup D_t}\mathscr{L}_d(d_i,\hat{d}_i)\\ &+\frac{\lambda_{pgl}}{n_s}\sum_{x_i\in D_s}\mathscr{L}_{\text{pgl}}(z_i,\hat{z}_i)\end{aligned} \tag{6.14}$$

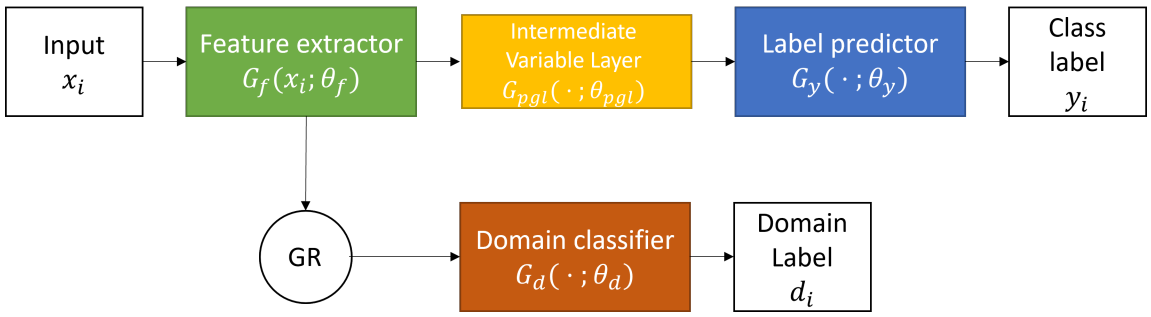


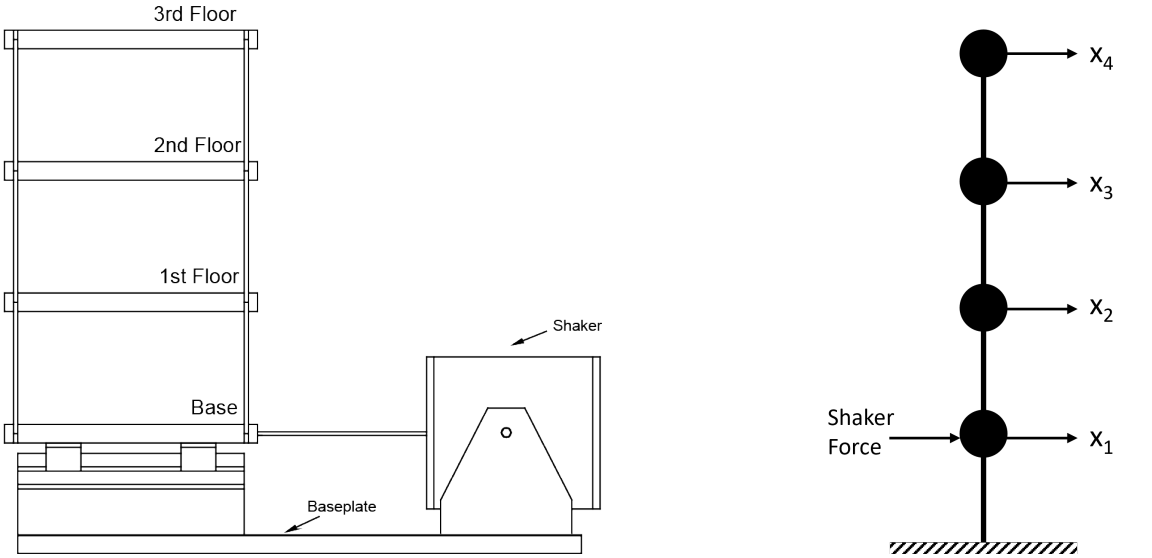
Figure 6.3: Concept of Physics-guided Learning

6.4 Implementation Details

6.4.1 Structure and Experimental Data

This chapter considers a small-scale three-story structure tested by Figueiredo et al. (2009) at the Los Alamos National Laboratory (see Figure 6.4b). Each floor made of aluminum plate is idealized as lumped mass and supported by four aluminum columns. The dimensions for the floor plates and columns are $17.7 \times 2.5 \times 0.6$

cm and $30.5 \times 30.5 \times 2.5$ cm, respectively. Columns are attached to the plates using bolted joints representing rigid connections. An electro-magnetic shaker attached to a baseplate excites the structure with white noise to generate structural responses. The vibration responses are captured by accelerometers at each floor including the base with a sampling rate of 320 Hz for a duration of 25.6 sec while the structure is excited with the white noise by the shaker. The excitation is band-limited within the range of 20-150 Hz to avoid rigid body modes. Sun and Betti (2015) identified three natural frequencies of the structure as $[31.09, 55.05, 72.23]$ Hz and a body rigid body motion mode.



(a) Experimental Setup (Figueiredo et al., 2009)

(b) Lumped model (Sun and Betti, 2015)

Figure 6.4: Three-story Los Alamos Laboratory structure

Various damage conditions are simulated by replacing the original column members with weaker members. In the original dataset, seven damage conditions (including the undamaged condition) are considered where the section of one or two columns at each floor is reduced to simulate damage (see Table 6.1). For each damage condition, 50 experimental trials are conducted. The duration of each trial is 25.6 seconds. This research focuses on localizing the cumulative column damage for each floor rather than by individual columns. This dataset is designated as *the modified dataset*. Any damage occurring at a particular floor is considered as one damage condition. As a result of this, the labels are condensed into four damage conditions including the undamaged condition.

After this preprocessing, all data are split into 1 second data segments and each segment has 320 samples per channel. In typical SHM applications, base vibrations are usually used only for modal analysis. We followed this practice and discarded acceleration measurements taken from the base sensor. The data chunks from the remaining three floors are stacked and paired with a label corresponding a damage location to form

Table 6.1: Damage conditions for original and modified dataset

Original Dataset		**Modified Dataset**	
Label	State Condition	Label	State Condition
0	No Damage	0	
1	First Floor Single Column 50% stiffness reduction each	1	First Floor Column Damage
2	First Floor Two Columns 50% stiffness reduction each		
3	Second Floor Single Column 50% stiffness reduction each	2	Second Floor Column Damage
4	Second Floor Two Columns 50% stiffness reduction each		
5	Third Floor Single Column 50% stiffness reduction each	3	Third Floor Column Damage
6	Third Floor Two Columns 50% stiffness reduction each		

an input-output pair.

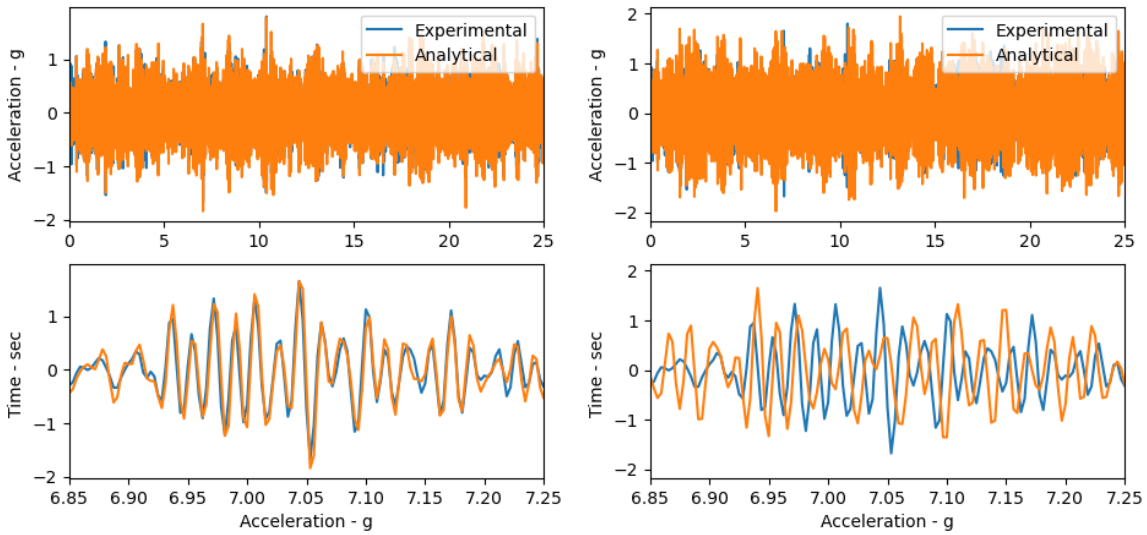
Next, the data is split into a training and testing set with a proportion of 3:1, respectively. Furthermore, the data is standardized by removing the mean and scaling to unit variance. The resulting data constitutes the training and testing target domain data.

6.4.2 Simulation Model and Analytical Data

A high fidelity lumped mass model of the structure is generated based on the prescription provided by Sun and Betti (2015) in the form of mass-damping-stiffness matrices (see Figure 6.4a). This model is formed as 4-DOF lumped mass system where the base is regarded as the first floor. In addition, a copy of this model is created with intentional misrepresentation to incorporate modeling error that is typically introduced in the design process. In particular, the modeling is introduced into the stiffness matrix by reducing the cumulative stiffness of each floor by 10 percent. It should be noted that even the high-fidelity model with no modeling error has some inaccurate representation due to inherent modeling uncertainty such as geometric and material nonlinearities. For this study, inherent uncertainty is ignored.

The quality of the model used for the source data generation is illustrated in Figure 6.5. Here, the experimental and analytical top floor accelerations are compared for the no damage condition of the structure under 0 and 10 percent modeling error. For both plots, the same input excitation is used. When no modeling error is present, the discrepancy between analytical and experimental data is minimal (see Figure 6.5a). The response comparisons are consistent with the coefficients of variation reported by Sun and Betti (2015) which vary between 0.4% to 1.0% for the lumped mass and floor stiffness values. As for the 10 percent modeling

error case, the difference between responses shown in Figure 6.5b) demonstrates the dataset shift is prominent between experimental and analytical data.



(a) No modeling error

(b) 10 percent modeling error

Figure 6.5: Comparison of experimental and analytical top floor acceleration responses

For each model (perfect and misrepresented models), we generate the response and the damage labels using simulation according to the damage conditions prescribed for the modified dataset in Table 6.1. After the data is segmented into 1-second segments, it is partitioned into datasets for training, validation, and testing with a proportion of $(3 : 1 : 1.33)$, respectively. The ratio of training to validation is $3 : 1$ and the ratio of training+validation to testing is $3 : 1$. Next, the data is standardized with respect to training data. All the data generated by the simulation model constitutes the source domain data.

6.4.3 Further processing DA, PGL, and DAPGL

A typical DANN require a domain classifier to discriminate the input data based on the origin of domain. To accommodate this classifier, a new label indicating the domain origin (0 for source; 1 for target) is generated for both domains.

Our PGL and DAPGL approaches discussed in this chapter require intermediate physics-based parameters. The intermediate variables selected for this study are the modal parameters (natural frequencies and mode shapes). These parameters are directly derived from the mass and stiffness matrices forming the simulation model through Eigensystem Analysis. Each damage condition has a unique mass and stiffness matrix pair. Thus, the modal parameters are specific to the damage condition. In addition, the modal parameters are identical for all samples within the damage condition class as all samples are generated with same stiffness

and mass matrix pair.

Only the first three natural frequencies and mode shapes are considered and modes relevant to rigid body motions are omitted. Higher order modes have lower energy content, thus, they do not contain high-quality features and are ignored. Based on the setup provided in Figure 6.4a, the structure is assumed to have 4 floors, i.e. the model shape will have 4 modal points per natural frequency. Since only three natural frequencies are reported, there are in total 3 modes x 4 DOF = 12 mode shape points. To sum up, three natural frequencies and 12 mode shape points [3 modes × 4 floors] are extracted from the simulation model for each damage condition and vectorized.

6.4.4 Implementation

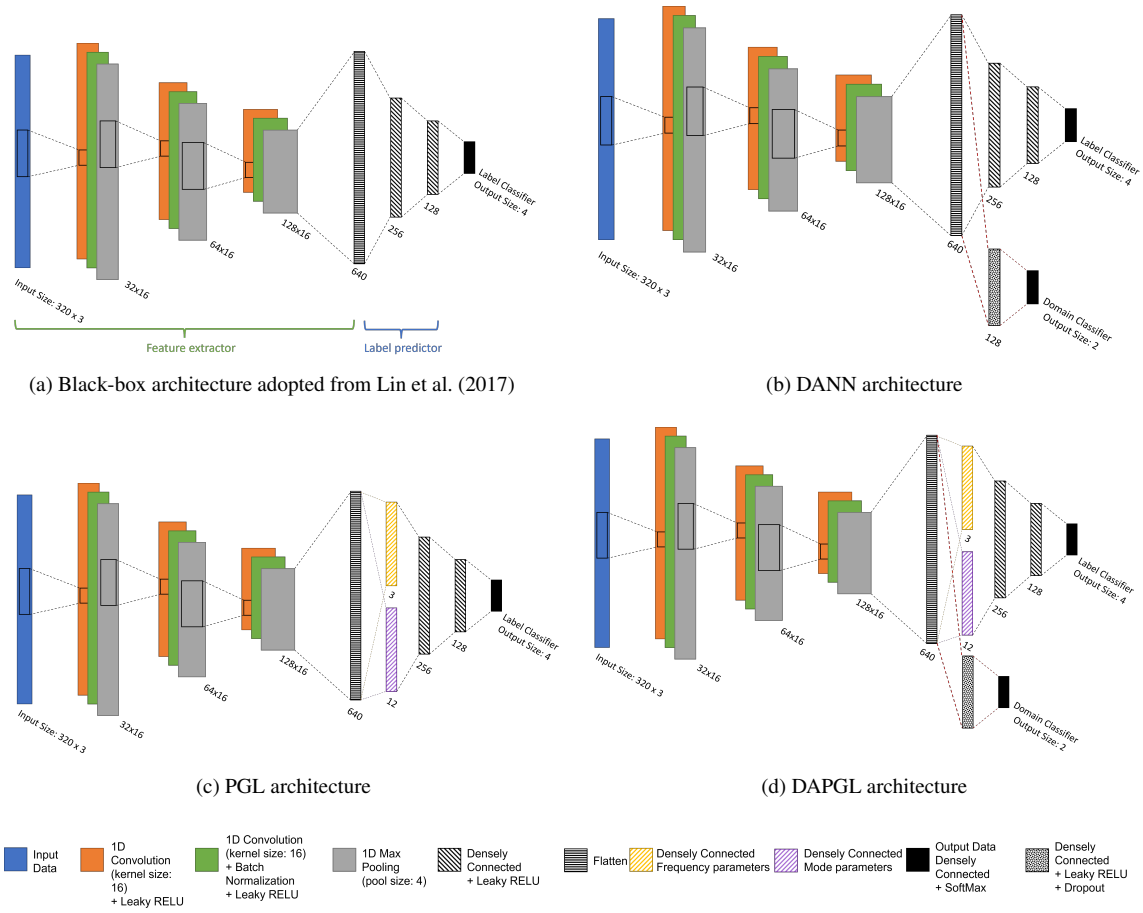
i. Black-box Model: A black-box neural network serves as the baseline and learns a function that relates the time series input with the damage location without the ingestion of physics-based intermediate variables or adversarial domain adaptation. The subsequent architectures listed below is based on this reference model.

The architecture for this network is prescribed Lin et al. (2017). The input for the black-box architecture is time-series data from top three floor sensors. The feature extractor component is composed of a set of 1D convolutional layers. Each CNN layer is followed by a batch normalization layer in order to mitigate the internal covariate shift. Lastly, to help over-fitting and reduce the number of trainable parameters to learn, a max pool layer is added after every batch normalization layer. The subsequent component is a set of densely-connected layers for predicting the labels. All neurons within these two components have leaky RELU activation functions. Lastly, the output layer is the damage location to be predicted and uses a softmax activation function. The blackbox architecture is illustrated in Figure 6.6a. The number of output filters and the length of the convolution window, as well as the number of neurons in densely-connected NN layers are provided in the figure.

This model is trained only with labeled source domain data. Unlabeled target domain data is utilized during the testing time.

ii. DANN Model: To understand the effect of domain adaptation on the performance, this research considers a deep learning architecture where a domain classifier is integrated into the black-box model. The DANN architecture used for evaluation is illustrated in Figure 6.6b. This model utilizes the loss function given in Equation 6.3. It should be noted that this model is trained both with labeled source domain and unlabeled target domain data.

iii. PGL Model: To understand the effect of physics-guided learning on the performance, a deep learning architecture is considered where the physics-based parameters are integrated into the black-box model as part of the learning process. The PGL architecture used for evaluation is illustrated in Figure 6.6c. This model



(a) Black-box architecture adopted from Lin et al. (2017)

(b) DANN architecture

(c) PGL architecture

(d) DAPGL architecture

Figure 6.6: Architectures used for evaluation

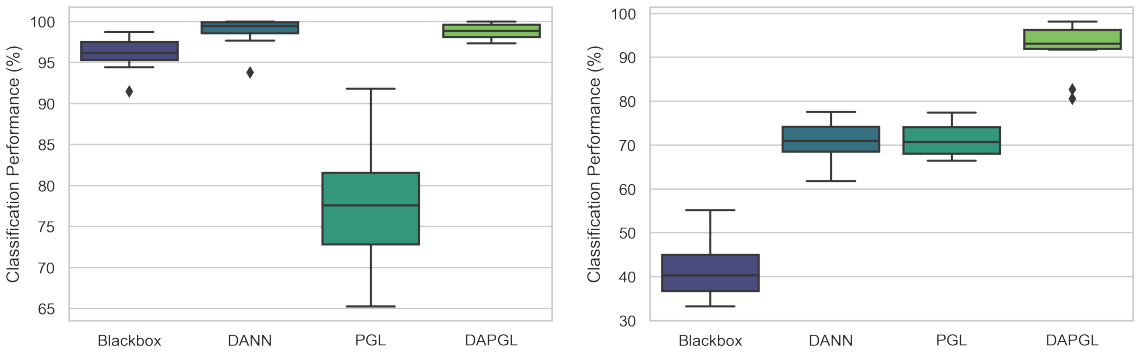
utilizes the loss function given in Equation 6.9. The intermediate variables selected for this model are natural frequencies and mode shapes. This model is trained only with labeled source domain data. Unlabeled target domain data is utilized during the testing time.

iv. DAPGL Model: Our proposed framework combines domain adaptation with physics-guided learning. The DAPGL architecture used for evaluation is illustrated in Figure 6.6d. This model utilizes the loss function provided in Equation 6.14 and it is trained both with labeled source domain and unlabeled target domain data.

Based on the previous studies (Ganin et al., 2016; Ozdagli and Koutsoukos, 2021b), the trade-off parameters for all the models are chosen such that losses are balanced equally. For all of the models, in total, labeled 5000 samples from the source dataset are available. The number of samples is divided equally among 4 classes. The samples are portioned across training, validation, and testing data as [2812, 938, 1250], respectively. Additionally, for DANN and DAPGL, unlabeled 3750 data points are available from the target dataset. The samples are portioned across training, and testing data as [2812, 938], respectively. All models use the testing data from the target domain dataset for evaluation.

6.5 Results

Before evaluating the DAPGL, we first generated the training source domain dataset using two simulation models with different modeling errors (ME). For each architectures discussed in the previous section, 10 models are trained with source domain dataset and tested against the experimental target test dataset. The classification performance box-plots of all models for different modeling errors are illustrated in Figure 6.7.



(a) ME 0%

(b) ME 10%

Figure 6.7: Classification performance under model uncertainty

When the modeling error is zero, the blackbox model has a mean performance of around 96% with small spread. DANN model has a small improvement over blackbox model. This small improvement can be attributed to the fact that even the perfect simulation model has some modeling uncertainty. Compared to blackbox and DANN, PGL model has the worse accuracy and largest spread despite its good performance

over source test data (99.46 percent - not shown in figures). The intermediate variable layers have in total 15 neurons and the flattening layer prior has 640 neurons. The decrease in performance could be attributed to the severe depression in number of neurons due to the implementation approach of the PGL model. This layer layout could hinder generalization capability to some extend. Among all of the architectures, DAPGL has a very minimal improvement over DANN. The results so far demonstrate that our framework generalize well for the training data when the modeling uncertainty is negligible.

When the modeling error is 10 percent, the blackbox model experiences a severe decrease in classification performance. On the other hand, DANN generalizes over the experimental data reaching to an average performance of 71%. Similarly, PGL has also a classification performance of 71%. This implies that the physics based intermediate parameters generalize the learning experience over experimental data well under the learning under modeling error despite the depression in neuron numbers at the intermediate layers. Among all of the architectures, DAPGL has the best classification performance by far. Results demonstrate that combining domain adaptation with physics-guided learning improve the generalization.

Table 6.2: Improvement of DAPGL over reference architectures (All values in percentage)

Modeling Error	Blackbox	DANN	PGL	DAPGL	Improvement
ME 0%	96.12	98.73	77.91	98.74	0.0
ME 10%	41.60	70.95	71.19	92.07	22.68

Table 6.2 presents the improvement of DAPGL over reference architectures. The improvement is computed by comparing DAPGL against the best performing model among the remaining three for a given modeling error case. While the improvement for ME 0% is negligible, the performance gain under modeling uncertainty/error reaches up to 22%. The significant increase in accuracy shows that intermediate layers combined with adversarial domain adaptation is capable of extracting a latent feature space that can generalize well over experiential data.

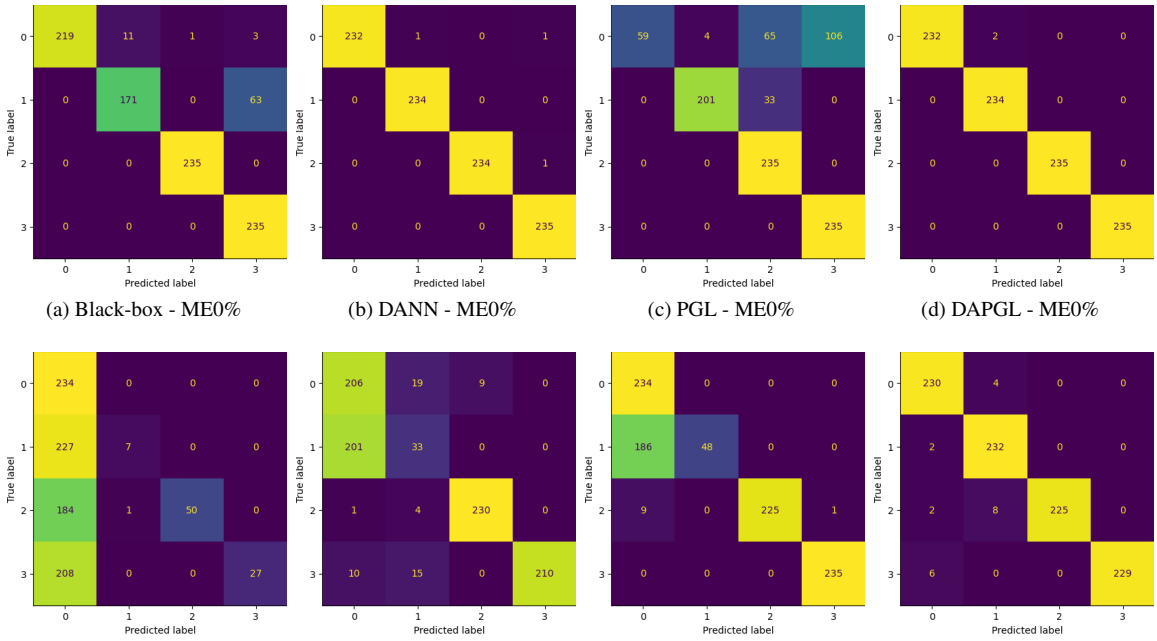
The classification performance for each damage condition under 0% and 10% modeling error is illustrated in Figure 6.8a. The confusion matrices are generated using a representative run from each model.

The blackbox model performs fairly at the class-level prediction for ME 0%. Its performance is slightly affected by inherent modeling error. The performance decrease for the blackbox model is very evident for ME10% case. The majority of wrong labels are concentrated in no damage condition. The natural frequencies of the misinformed numerical model are smaller than the experimental values since the modeling error introduces a 10% reduction of the stiffness on all floors. A network trained with such source data underestimates the damage condition since the target data used for testing has higher frequency content. As a result of that, the blackbox network regards the source domain data as *healthy*, even for higher damage conditions.

DANN has a very good performance for the ME 0% case since the domain adaptation from source to target can minimize the small probabilistic divergence between numerical and experimental data efficiently. However, its effectiveness diminishes under modeling error, mainly due to misclassification of the first floor damages. This observation implies that the domain knowledge transfer for the second and third-floor damage conditions is effective whereas DANN's capability in discriminating first-floor damages is unreliable.

As for PGL, the baseline model (ME 0%) is mislabeling mainly the no damage condition. While the bottleneck effect discussed previously affects the network performance for ME 0% case negatively, the generalization gained through PGL benefits the classification for the ME10% case greatly.

Among all networks, DAPGL has the best classification performance at the class level. DAPGL is as good as DANN for ME0%, whereas, under modeling uncertainty, DAPGL still has a successful prediction performance. Results show that ingestion of physics-based knowledge into the learning and domain adaptation improves the generalization capability of DAPGL.



(a) Black-box - ME0% (b) DANN - ME0% (c) PGL - ME0% (d) DAPGL - ME0%

(e) Black-box - ME10% (f) DANN - ME10% (g) PGL - ME10% (h) DAPGL - ME10%

Figure 6.8: Classification performance for each damage condition under model uncertainty

6.6 Conclusion

This chapter presented a deep learning framework for damage localization by combining physics-guided learning with domain adaptation. The performance of common ML-based SHM applications is often hindered by the lack of high quality training data or a shift in the structure behavior due to the changes it experiences during the life-cycle. By ingesting domain adaptation and physics-based domain specific knowledge into the

training procedure, our proposed framework can generalize the damage localization over target domain.

While the problem formulation considered in this chapter focuses on damage localization applications, the framework can be tailored to any SHM problem where dataset shift is observed between source and target domains and learning can be constrained through the ingestion of physics-based knowledge. In addition, our method promises a novel platform explainable through intermediate physics-related parameters. We believe that the improved interpretability and explainability can have a meaningful impact as the data produced by this framework is recognizable by humans.

6.7 Future Work

The current approach assumes that target domain data is completely unlabeled. In reality, a newly deployed structure is expected to be health as it has not experienced damage yet. As a result, data belonging to this state of the structure is in fact labeled as *undamaged*. In other words, labeled target domain data is partially accessible for training. This data contains valuable latent feature space information which can be used for improving generalization over both domains more effectively. For future research, approaches such as multi-adversarial domain adaptation by Pei et al. (2018) should exploit this labeled target domain data by enabling stronger alignment between target and source domains through multiple domain discriminators.

The classical adversarial domain adaptation assumes that task-relevant target domain data is available during training. In reality, for some cases, neither data sample nor label in the target domain may be accessible. For a damage classification task, adversarial training only with the readily available no damage condition data from the intact structure, i.e. single class from the target domain is not appropriate, as the task is often a multi-class problem. Domain adaptation is still a maturing area, and the community is well aware of this observation. To remedy the aforementioned limitation, a new branch of transfer learning method called Zero-Shot Domain Adaptation has gained interest which does not require target data for training (Kodirov et al., 2015; Peng et al., 2018; Wang and Jiang, 2019). The future research should investigate the applicability of zero-shot training along with physics-guided learning toward SHM problems.

Another future research direction is the exploitation of explainability through DAPGL. In addition to intermediate layers, methods such as Layer-wise Relevance Propagation (Bach et al., 2015) will enable the verification of reasoning. Lastly, future research should address the performance degradation in PGL due to bottleneck for the cases where a small-capacity physics-guided intermediate layer is followed by a large layer. The preliminary results imply that augmentation of the intermediate layer into a previous or subsequent layer may improve the prediction quality.

This research uses a relatively traditional architecture for damage classification. The use of new classification approaches such as Neural Dynamic Classification Algorithm (Rafiei and Adeli, 2017) and Dynamic

Ensemble Learning Algorithm (Alam et al., 2020) should be explored in conjunction with DAPGL for further prediction accuracy improvement.

CHAPTER 7

Interpretability of Neural Network Models through PGL

7.1 Introduction

While an appropriately trained black-box machine learning algorithm is capable of generalizing well over the testing data, model's inference process and its predictions are often difficult to interpret especially for deep-learning architectures employing datasets with high nonlinearities. The black-box model can learn the nonlinearities within the dataset with a high accuracy. However, the underlying reasons suggesting the decision made by the model is often obscure to the end user due to the transparent nature of the architecture (Bhatt et al., 2020). The gap in the knowledge caused by the lack of interpretation may hinder stakeholders in analyzing the results in depth and debugging the model.

To facilitate an interaction between the user and the model, one could map the high-dimensional model behavior to the real-world phenomena surrounding the task and create *glass-box* approaches (Holzinger, 2018). Roscher et al. (2020) states incorporating domain knowledge into machine learning could provide and enhance the transparency, interpretability, and explainability. Through domain knowledge, experts such as data scientists, business owners, risk analysts, regulators, as well as consumers could greatly benefit from trustable and consistent decisions and add further value to the decisions made by the machine learning (Belle and Papantonis, 2021).

The first attempts in the development of explainable AI methods started in 2015 with layer-wise relevance propagation (LRP) technique (Bach et al., 2015). LRP is an approach developed for explaining neural networks whose inputs are images, videos, or text. It computes the back-propagation of gradients from the prediction to the input using a specific set of propagation rules and highlights the input feature that is most relevant to the output. Another method called local interpretable model-agnostic explanations (LIME) proposed by Ribeiro et al. (2016) focuses on a more model-agnostic approach by approximating the underlying model through an interpretable linear model which is trained on the perturbations of the input. The majority of the subsequent explainable AI methods has followed the main two categories posed by LRP and LIME, neural-network based and model-agnostic approaches, receptively.

In general, the neural-network based approaches follow LRP-like approach and adopts backpropagation of gradients to explain results (Selvaraju et al., 2017; Kim et al., 2018; Sundararajan et al., 2017). Those applications are mostly focusing on visual and textual tasks. On the other hand, the model agnostic approaches are more flexible in terms of applications. A prominent method called SHapley Additive exPlanations (SHAP)

presents a unified framework for interpreting predictions (Lundberg and Lee, 2017). Since SHAP built a well understood foundation for explaining models, a series of literature adopted this method (Frye et al., 2020; Antwarg et al., 2019). Ribeiro et al. (2018) proposed Anchors to explain the behavior of complex models with high-precision rules, There are also novel model-agnostic approaches for explaining graph-like structures such as LIME based GraphLIME (Huang et al., 2022), SHAP based Shapley Flow (Wang et al., 2021), and GNNExplainer (Ying et al., 2019).

To the author's best knowledge, the majority of the literature mentioned above do not directly incorporate the domain knowledge into the explanation. The aforementioned methods are mainly focused on creating a glass-box transparent environment that relates the prediction to the input by highlighting the feature importance. While these explanation methods significantly help the end user to establish an interpretation of the results, the knowledge obtained is limited to the input dataset and expertise of the user. In this section, we propose an interpretable physics-guided learning approach that utilizes intermediate domain-specific knowledge with two goals in mind: (i) The existing methods do not consider time series data in explaining the results. This work demonstrates the explainability is applicable for time-series data. (ii) The domain knowledge incorporated into the explainable AI is mostly limited by the input features. This work seeks to extend the domain knowledge through intermediate physics-related parameters.

This chapter first introduces a brief discussion on layer-wise relevance propagation (LRP), prorogation rules typically used in deep networks, and some details towards effective implementation. Then, the section briefly discusses the PGL architecture used for this study, and how LRP is used in conjunction with this architecture. Following the methodology setup, a case study considered to understand the effectiveness of LRP in interpreting the PGL network and the details are discussed. Finally, conclusions are summarized.

7.2 Methodology

7.2.1 Layer-Wise Relevance Propagation

This work adopts layer-wise relevance propagation (LRP) work proposed by Bach et al. (2015). The propagation procedure of LRP employs propagating relevance score (R_k) for a given layer to a lower layer as given below:

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k \tag{7.1}$$

where j and k denotes the neurons at two subsequent layers; z_{jk} is the contribution of relevance for the neuron j towards neuron k. Eq. 7.1 implies summation of relevance scores within a layer for all neurons is conserved throughout the network, i.e. $\sum_j R_j = \sum_k R_k$.

7.2.2 LRP Rules

Assuming LRP is applied to deep networks with rectifiers, the rectified neuron behavior can be defined as:

$$a_k = max(0, \sum_{0,j} a_j w_{jk}) \tag{7.2}$$

Accordingly, the basic LRP rule, also known as LRP-0 can be defined as:

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k \tag{7.3}$$

LRP-0 tends to pick local artifacts frequently. Thus, as an extension to this rule, to create more sparser explanations, a small term ε can be added (see Eq. 7.4). As ε becomes larger, the contribution of weak and contradictory neurons can be absorbed. This rule is also known as LRP-ε.

$$R_j = \sum_k \frac{a_j w_{jk}}{\varepsilon + \sum_{0,j} a_j w_{jk}} R_k \tag{7.4}$$

Finally, LRP-γ rule introduces the parameter, γ to favor the effect of positive contributions over negative ones as given below:

$$R_j = \sum_k \frac{a_j (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j (w_{jk} + \gamma w_{jk}^+)} R_k \tag{7.5}$$

Here, when the term γ increases, negative contributions to the relevance start to disappear. Montavon et al. (2019) states that LRP-γ limits the growth of positive and negative relevance by prevailing the positive contributions which leads to more stable and understandable explanations.

Since the first layer of the architecture are often formed as real values, a special LRP rule called LRP-w^2 is used as given below:

$$R_i = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_k \tag{7.6}$$

7.2.3 General Implementation

The three rules, LRP-0, LRP-ε, and LRP-γ can be implemented through a generic rule such that:

$$R_j = \sum_k \frac{a_j \rho(w_{jk})}{\varepsilon + \sum_{0,j} a_j \rho(w_{jk})} R_k \tag{7.7}$$

where the function, ρ determines which LRP rule to follow. The generic propagation rule can be decomposed into four steps as given below:

$$z_k = \varepsilon + \sum_{0,j} a_j \rho(w_{jk}) \tag{7.8}$$

$$z_k = R_k / z_k \tag{7.9}$$

$$c_j = \sum_k \rho(w_{jk}) s_k \tag{7.10}$$

$$R_j = a_j c_j \tag{7.11}$$

7.2.4 Rule Selection based on Layer Depth

In general, different LRP rules are imposed for various depth of layers. Here, we describe the depth of the layer as upper, middle, and lower based on its distance to the input; upper layer being furthest. In typical applications, LRP is computed starting from the output and backprogates towards the input. Commonly, upper layers contain large amount of neurons per class which leads to a entanglement of information. For these layers, LRP-0 is used since it is known to be insensitive to the entanglement. For middle layers, while the representations start to disentangle, the weight sharing among the convoluted layers can cause noise in the explanation. Thus, LRP-ε is a more suitable fit as it filters the noise and prevails the most salient explanations. Lastly, for lower layers, LRP-γ is used to deliver a more human understandable explanation. More specifically, LRP-γ delivers a more stable and uniform explanation instead of spreading the minor contributions spuriously among features.

7.3 PGL Architecture

In the previous two sections, PGL architecture is thoroughly discussed. This section follows the architecture prescribed in Figure 5.1. In this study, the input for this architecture is the time domain data, output is the location of the damage, and the intermediate layer values are selected as modal paramaters such as natural frequencies and mode shapes. Here, the purpose of LRP for this architecture is attributing the predicted damage classes to the modal parameters and observe which parameters are most dominant for the decision the PGL came up with.

The list of layers and the LRP rules used are tabulated in Table 7.1. For the upper layers which consist of linear layers (layer count 24-30), the LRP-0 is selected. LRP-γ rule governs the propagation for the mid layers (layer count 9-22) which are composed of stacked convolutions. Finally, for the lower layer (layer count 2-8), the architecture utilizes the LRP-ε rule. For this application, parameters are $\varepsilon = 0.25$ and $\gamma = 0.25$. The softmax layer is excluded from LRP propagation since the top layer may not be always selective towards

class explanation especially for small amounts of damage. Since the input layer consists real time-domain values, the special rule, LRP-w^2 is applied. For propagation purposes, LeakyReLU activations are considered as regular ReLU. Since they are absorbed by their preceding layer through the propagation, no special LRP rule is applied. Special layers such as MaxPool follow the LRP rules prescribed in Table 7.1. The batch normalization layer acts merely as a centering and scaling operation during the testing time, thus LRP rule does not apply. Likewise, the dropout layer is also ignored during testing time and no special rules are applied.

Table 7.1: LRP layer rules

Layer Count	Layer Type	LRP Rule
1	Input	LRP-w^2
2	Conv1d	LRP-γ
3	LeakyReLU	-
4	Conv1d	LRP-γ
5	BatchNorm1d	-
6	LeakyReLU	LRP-γ
7	Dropout	-
8	MaxPool1d	LRP-γ
9	Conv1d	LRP-ε
10	LeakyReLU	-
11	Conv1d	LRP-ε
12	BatchNorm1d	-
13	LeakyReLU	-
14	Dropout	-
15	MaxPool1d	LRP-ε
16	Conv1d	LRP-ε
17	LeakyReLU	-
18	Conv1d	LRP-ε
19	BatchNorm1d	-
20	LeakyReLU	-
21	Dropout	-
22	MaxPool1d	LRP-ε
23	Flatten	-
24	Linear - Intermediate Layer	LRP-0
25	LeakyReLU	-
26	Linear	LRP-0
27	LeakyReLU	-
28	Linear	LRP-0
29	LeakyReLU	-
30	Linear	LRP-0
31	Softmax	-

7.4 Implementation

For this case study, the simply supported beam discussed in Section 3.3.2 is used as the template to generate the training data. The parameters selected for the beam model is listed in Table 7.2. Using the section and material properties, a FE model is constructed in OpenSees, where the beam is discretized into 10 beam elements.

In addition to the weight of the beam, the nodal mass is added to each node such that the first natural frequency matches 2 Hz. OpenSees reported the first five natural frequencies as $f = [2.00, 8.01, 18.01, 31.97, 49.71]$ Hz. The beam is excited vertically at the support level with a band limited white noise to simulate the ambient vibrations. The length of a typical simulation is 10 seconds and its sampling frequency is 200 Hz. Vertical structural responses to the given excitation are collected at 9 nodes as time domain data, excluding the support nodes. In addition to the healthy condition of the beam, a damage class is designated for each discretized element. The damage is introduced to the element by reducing the elastic modulus by 20%. Including the no damage case, there are 11 damage cases. In total, 1100 samples are generated through OpenSees where each damage class has 100 samples. Each sample has a dimension of [9 nodes $\times$ 2000 time data point]. Additionally, for every sample, corresponding modal parameters composed of five natural frequencies and mode shapes for the first three modes are obtained analytically. This data poses as intermediate physics-guided parameters and its dimension is [5 natural frequencies + 3 modes $\times$ 9 modal points per mode = 32 data point].

Table 7.2: Beam model properties

Property	Acronym	Value
beam length	L	5 m
Section width	b	0.1 m
Section height	h	0.25 m
Elastic Modulus	E	206 GPa
Material Density	ρ	7900 kg/m^3
No of elements	e	10
Nodal mass per node	m	13000 kg

Both time domain data, physics-guided parameters, and the corresponding damage classes are divided into training and testing data in the ratio of 4:1. The loss functions associated with class and physics-guided parameters have equal weights for training. The network is trained with PyTorch using gradient decent in 3000 epochs and the testing accuracy yields 100% correct class prediction. After the network training finalized, LRP rules shown in Table 7.1 are applied.

7.5 Results

Figure 7.1 presents the LRP intensities in time domain for various damage cases. There are 9 time series for each plot. Each time series corresponds to a structural response captured at a node as labeled in the plot. The color illustrates the relevancy of the predicted class to the responses. Here, red and blue colors describe high and low relevancy, respectively, whereas gray color indicated no relevancy. While LRP provides a detailed relevancy map in terms of time series, the explainability is still very limited to the human understanding.

Therefore, we looked into the intermediate physics-guided parameters that are relatively easier humans to understand.

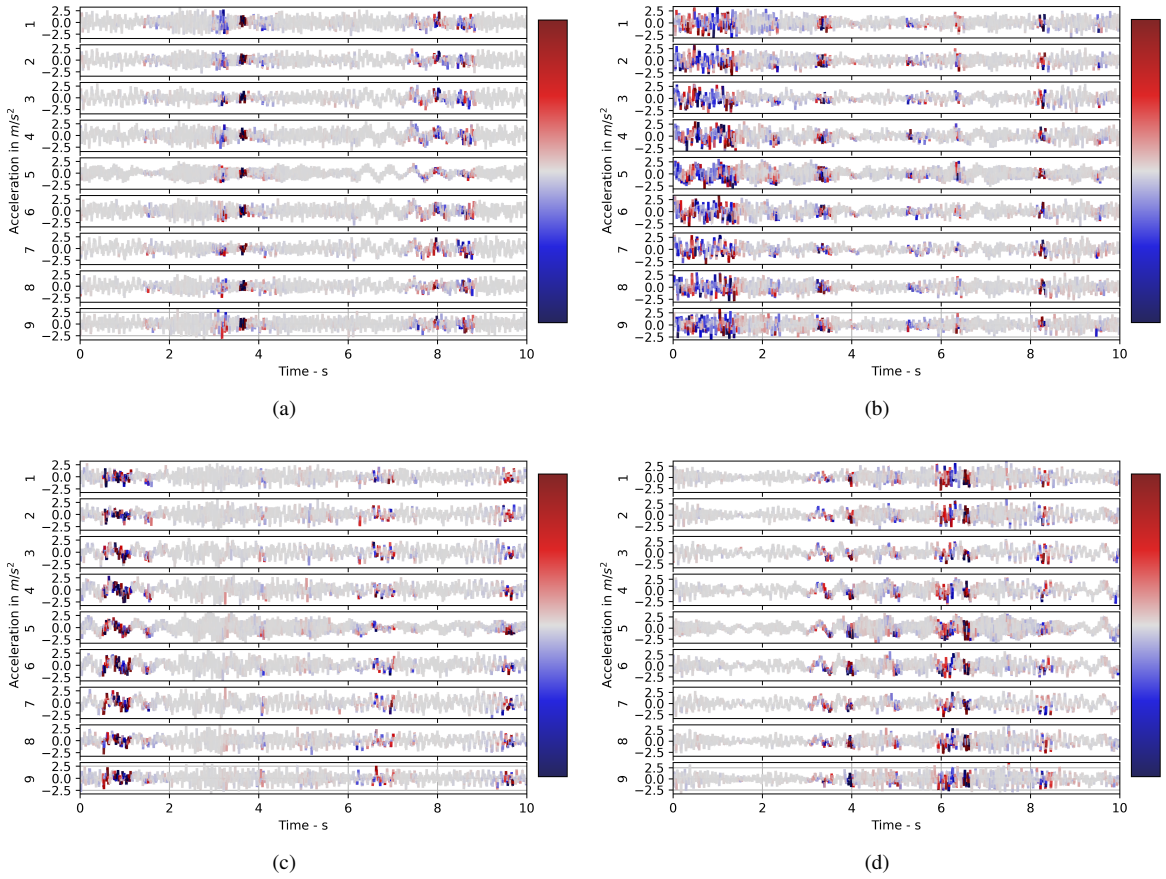


Figure 7.1: Interpretability in time domain for various damage cases: (a) no damage; (b) damage @ member 1 - left side of the beam; (c) damage @ member 5 - midspan; (d) damage @ member 10 - right side of the beam

Figure 7.2 illustrates the modal parameters for the first three modes for various damage cases. The results present both true and predicted mode shapes for each mode and their natural frequencies. Here, to the naked eye, the natural frequencies are the only dominant discriminatory intermediate parameters. For example, the first natural frequency of the undamaged system is 2.0028 Hz, whereas the case where the member 5 is damaged has a natural frequency of 1.9560 Hz. As for the mode shapes, beam shapes are visually identical for the first three modes. Accordingly, without further statistical analysis, the intermediate physics-related parameters are by themselves qualitatively not valuable.

Figure 7.3 displays the LRP properties of intermediate physics-based parameters, more specifically the first mode shape values. Here, the progression of LRP relevances clearly demonstrates that the LRP score highlights the members that are most likely have the damage for the given structural responses captured from

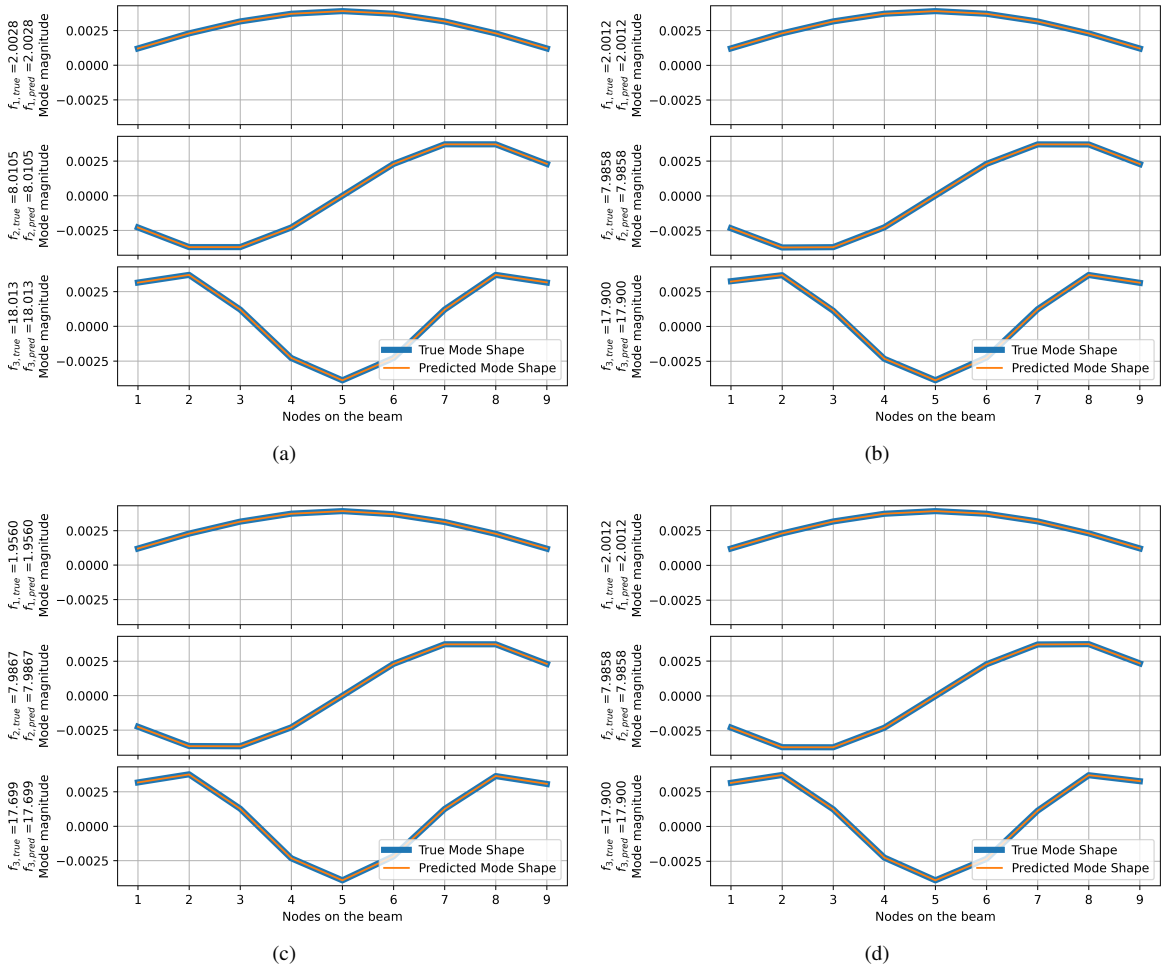


Figure 7.2: Comparison of physics-based parameters for various damage cases: (a) no damage; (b) damage @ member 1 - left side of the beam; (c) damage @ member 5 - midspan; (d) damage @ member 10 - right side of the beam

the beam. For instance, according to the Figure 7.3(a), the LRP of the mode shape indicates that there is a damage concentrated around the first beam element for a case predicted as *damage @ member 1*. Similarly, for damaged midspan elements (see Figure 7.3(e and d)), LRP emphasizes on the mode shape values at midspan and points out the most relevant region for the prediction. As observed here, small changes in actual mode shape values are not recognizable by a human as a discriminatory feature for explaining the prediction. However, LRP explanation on the first mode shape significantly improves the human interpretability and understanding in rationalizing the model prediction.

7.6 Conclusion

This chapter presented a method to explain deep learning architectures with embedded physics-guided parameters through layer-wise relevance propagation. The interpretability of a machine learning algorithms is often limited due to its black-box nature. Through explainable AI, one can improve the interpretability of the ML and can have a better understanding of its reasoning. In this study, we focused on improving the logic behind a prediction by relating the explanations to the intermediate physics related parameters specific to the domain application.

The method used in this study, LRP allows us to construct a relevancy map from the predicted output to the given input as well as the intermediate physics-based layers. As discussed in the chapter, LRP does not add additional value to the interpretation of the input which was recorded in time series data. While the architecture can detect and learn the small changes in physics-guided parameters to predict an accurate outcome, the parameters predicted by the algorithm may not be very discriminatory to human recognition without further analysis. We demonstrated that by using LRP, the predicted damage condition can be attributed to the damaged element through intermediate parameters such as the first mode shape.

In essence, the work presented here also show that a very accurate deep learning architecture that acts like a digital twin of the actual system could explain the complex system behavior through intermediate physical quantities. We believe that this property of explainable AI not only impacts the interpretability and explainability of the ML algorithm, but also it may aid humans to understand and construct new relationships between physical quantities through explainable AI.

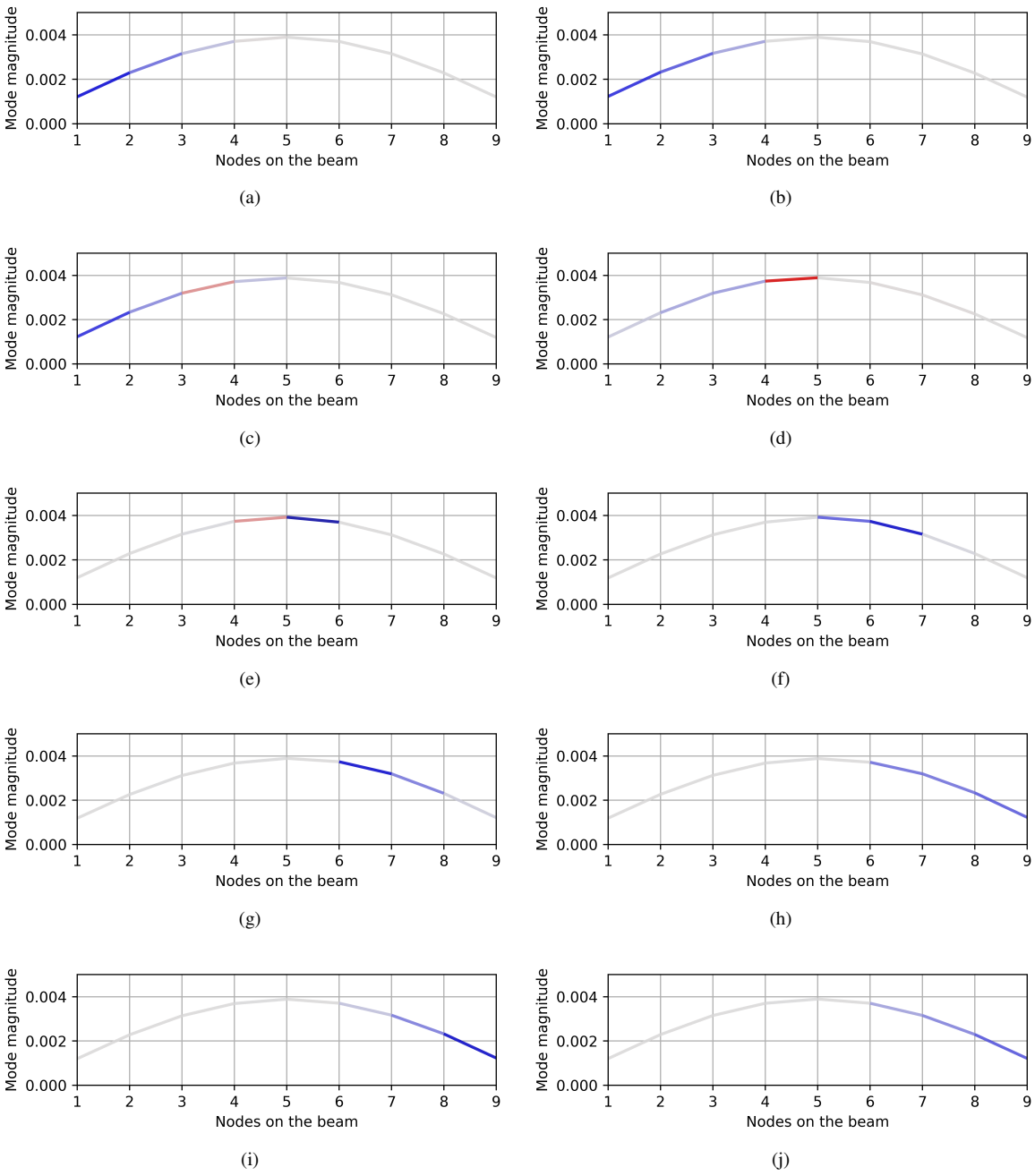


Figure 7.3: LRP for physics-based parameters for all damage cases: (a) damage @ member 1 - left side of the beam; (b) damage @ member 2; (c) damage @ member 3; (d) damage @ member 4; (e) damage @ member 5 - midspan; (f) damage @ member 6 - midspan; (g) damage @ member 7; (h) damage @ member 8; (i) damage @ member 9; (i) damage @ member 10 - right side of the beam

CHAPTER 8

Surrogate Modeling through PGL

8.1 Introduction

In the last few decades, computer simulation models gained an extremely high interest in scientific and engineering community as they are known to accelerate complex tasks and decision processes reliably. While through model calibration and parametric sensitivity analysis, one can design realistic and complicated simulations procedures with very high fidelity, such simulations are also known to be computationally exhaustive. Especially for modern model-integrated design environments involving multiple inter-disciplinary domain knowledge, challenging engineering problems require running long lasting simulations repeatedly which cost time and money to the stakeholders (Queipo et al., 2005).

Surrogate modeling, also known as metamodeling (Emerson and Sztipanovits, 2006) and sometimes digital twin (Jones et al., 2020), focuses on employing computationally less exhaustive surrogates of the actual model. In essence, the main motivation for surrogates is effective utilization of limited computational resources. The surrogate models often use a data-driven approach to approximate the behavior of the actual model. In recent year, neural network based surrogate models got accepted as a viable method since neureal networks are capable of learning complex nonlinear intrinsic relationship between given input and model response at a relatively low computation cost (Shrestha et al., 2009; Sreekanth and Datta, 2010; Papadopoulos et al., 2018).

A majority of NN-based surrogate modeling methods usually applies a data-driven black-box modeling where the inference procedure is often non-transparent to the designer. While the black-box techniques are very efficient and accurate as surrogates in predicting the responses, they may often fail in generalizing over the less explored design areas. Some novel applications (Daw et al., 2017; Zhang et al., 2020) implemented physics-guided learning into the training by customizing the loss function with the aim to improve generalization. However, these applications are still not transparent enough to give an explanation for the user exploring the design space.

This aim of this chapter is twofold. Firstly, this study proposes a physics-guided learning (PGL) approach where the deep learning architecture is infused with intermediate physics-related parameters as explained in Chapter 5. The purpose of this architecture is generalizing the surrogate model over the less explored design space compared to black-box networks. Secondly, this study seeks to introduce and improve the explainability of the PGL through intermediate physics-related using layer-wise propagation (LRP). The goal

of this approach is aiding the designer to determine the dominant features in the given design and decide on new designs effectively based on the interpretation of the relevance.

This chapter first introduces a brief discussion on the problem definition and the goals of the surrogate model studied here. Then, the chapter briefly discusses the PGL architecture used for the training of the surrogate model and the LRP approach used for the explanation of the architecture in terms of design and intermediate simulation parameters. Next, the surrogate model is evaluated in terms of generalization and explainability of the model is explored through a variety of designs. Finally, a summary of results are provided.

8.2 Problem Definition

In the last decades, autonomous vehicles have become a key research area and the growing body of literature suggests the autonomous systems are going to drastically change the future of transportation Mora et al. (2020), cyber-physical systems (Chen et al., 2017), and cyber-security (Chattopadhyay and Lam, 2017; Yağdereli et al., 2015) and warfare (Bruzzone et al., 2013; Hallaq et al., 2017). Another aspect of autonomous vehicles is the design of its cyber-physical components (Wilding, 2019). However, the complex interaction between different subsystems involving multi-disciplinary domain knowledge makes the design procedure extremely meticulous. Such designs often require repeated runs of high-fidelity computational simulations to optimize the final product (Cobb et al., 2022).

This study seeks to develop a surrogate model for simulating and analyzing the structural integrity of an unmanned underwater vehicles (UUV) hull (see Figure 8.1).

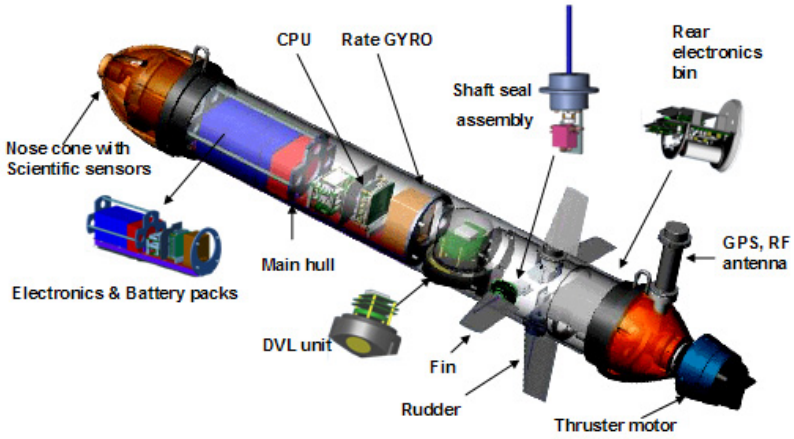


Figure 8.1: Hull example - adopted from Singh and Chowdhury (2011)

For a typical design, the shape of the hull depends on the placement and the size of systems like motor,

battery, and sensors. Each of these components should be optimized iteratively to improve the flight time of the UUV. Accordingly, at each iteration, the geometric properties of the hull needs to be redesigned to accommodate the component placements. Each redesign imposes a new hull configuration that should be verified through finite element simulations. For large problems, this patters implicates repeated runs of computationally exhaustive simulations which can slow down the optimization process. A neural network based surrogate model could reduce this process load significantly. Moreover, integration of physics guided knowledge into the training of the surrogate model and the introduction of the explainability through LRP could improve the generalization and interpretability of the model, respectively.

8.2.1 Research Objective

The ultimate aim is designing a physics-based learning - surrogate model architecture that generalizes the prediction and providing explainability based on the physical parameters relevant to the domain knowledge. To realize this goal, we need to come up with a design and data generation scheme and a design space exploration procedure. Figure 8.2 illustrates a typical surrogate modeling approach. Here, X is the set of design parameters that is drawn from the design space for the UUV. This set is used to design an experiment and generate data using the simulator. The output, Y is the desired result that is generated by the simulator. This data indicates how feasible a design is, based on the design parameters and the design requirements. For this study, the feasibility of a design is measured in terms of the internal stresses a UUV design is experiencing for a given depth. If the internal stress is close to the allowable stress limit imposed by material propoerty, this design has low feasibility.

After sufficient (X,Y) data pair is generated, the PGL can be trained with the following objective:

$$\min(Y - \hat{Y}) \text{ given } X \tag{8.1}$$

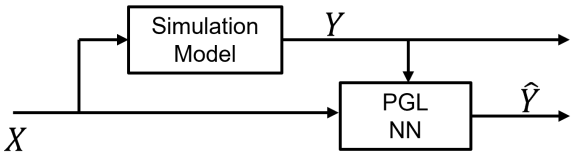


Figure 8.2: Typical approach for surrogate modeling

8.2.2 PGL-based Surrogate Model

In this study, we follow a PGL architecture discussed in the previous chapters for the surrogate model training (see Figure 8.3). Here, the input is the design parameters generated by oracle and the output is the feasibility ratio, which is the ratio between the allowable stress and maximum Von Mises stress the UUV design is experiencing. G_f (green) is the set of neural network layers that extract latent features from the given input and G_y (blue) is the set of neural network layers that generates feasibility regression. Without incorporating any physics-based parameters into the learning, i.e. treating the model as a black-box composition of G_f and G_y, the loss function to minimize would be the regression error between predicted and actual values:

$$\mathscr{L} = \frac{1}{n}\sum \mathscr{L}_y(y_i, \hat{y}_i) \tag{8.2}$$

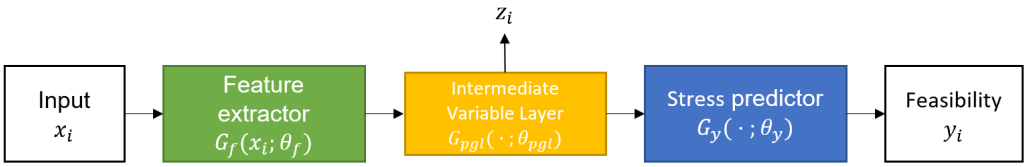


Figure 8.3: PGL-based Surrogate Model

As for physics guided learning, a new component called intermediate variable layer, G_{pgl} (yellow) is added to the architecture which is associated with physics-based parameter, z. This physics-based parameter can be also obtained through simulator. The loss function for the physics-based loss is described as:

$$\mathscr{L} = \frac{\lambda_{pgl}}{n}\sum \mathscr{L}_{\text{pgl}}(z_i, \hat{z}_i) \tag{8.3}$$

The final loss function for PGL is the aggregation of both loss functions discussed above.

$$\mathscr{L} = \frac{1}{n}\sum \mathscr{L}_y(y_i, \hat{y}_i) + \frac{\lambda_{pgl}}{n}\sum \mathscr{L}_{\text{pgl}}(z_i, \hat{z}_i) \tag{8.4}$$

8.2.3 Automatic Shape Generation

The first step for data generation is designing series of experiments. In this study, we follow a typical workflow illustrated in Figure 8.4. Suppose that, the oracle comes up with a design and the domain expert needs to establish if the design will remain structurally intact under water, i.e. domain expert needs to verify the design requirement. Based on the structural design parameters oracle produces, we can purpose a finite element model such as Ansys (ANSYS, 2022) to generate design shapes and simulate their response. Moreover, by

using a Python wrapper called PyAnsys (Kaszynski, 2020), one can automate the design generation and simulation. Eventually, any result obtained through simulations are backed up to oracle catalog towards building a constraint solver.

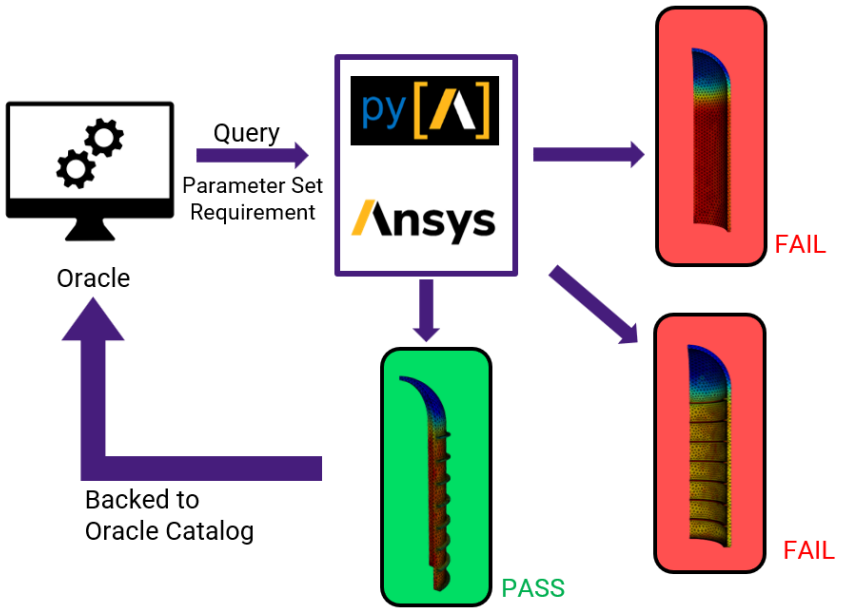


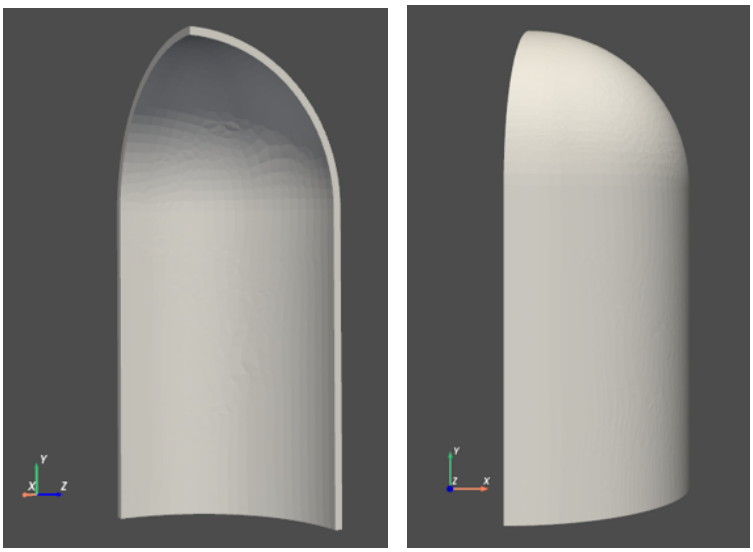
Figure 8.4: Automatic hull generation

However, running FEM simulations repeatably are usually costly. The oracle may have time budget and may not want to wait for the FEM simulator to finalize simulation and to get the results. Of course, the designer can come up with a blackbox surrogate model to predict responses fast and within a reasonable accuracy. On the other hand, the generalization can be sometimes problematic especially for the design space areas where there is not enough training data. Additionally, blackbox models don't provide any interpretability for explaining results. For such settings, PGL based surrogate models provide additional value.

8.2.4 Design Parameters

The UUV design we consider here is a simple cylindrical capsule model with spherical end caps (see Figure 8.5). The parameters relevant to the design are tabulated in Table 8.1 under two categories, oracle inputs, and simulator outputs.

The first four oracle inputs (grayed) are fixed by design, and they are related to the material properties. The following three parameters are geometric and governs the design of the hull shape. The next one, σ_{hyd} is related how much pressure will act on the vessel. This parameter acts also as a design requirement, such that the design is expected to sustain this pressure. The simulator generates multiple inputs. The last four parameters are special to capsule designs that maintain stiffener shape, number, and location. Examples for



(a) (b)

Figure 8.5: A representative capsule model generated with PyAnsys - 1/8 cut: (a) inside view; (b) outside view

inner and outside stiffened designs are illustrated in Figure 8.4.

For typical capacity-based design applications, the magnitude of maximum Von Mises stresses determines the feasibility of the design. This value is simplified into feasibility ratio where we only focus on the ratio between maximum Von Mises stress and the nominal material strength. There are also 9 additional internal stresses the simulator is capable of producing. We identified these parameters as the intermediate variables, z discussed in the previous section. The motivation comes from the fact that in order to compute the Von Mises stresses analytically from design inputs, one should be able to compute the other types of stresses first. In other words, there is an already established organic relationship between design inputs, intermediate variables, and stress. So this knowledge is introduced into the loss function using the intermediate variable stresses (see Eq. 8.5). Curious readers should consult Mises (1913) for more details.

$$\sigma_{\mathrm{v}} = \sqrt{\frac{1}{2}\left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2\right] + 3\left(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2\right)} \tag{8.5}$$

It should be noted that using the intermediate physics-related parameters, the loss function for PGL given in Eq. 8.4 can be redefined as given below:

$$\mathscr{L}_{pgl}(z_i, \hat{z}_i) = \lambda_{\mathrm{pgl},1}\mathscr{L}_{\mathrm{pgl},1}(\sigma_x, \hat{\sigma}_x) + \lambda_{\mathrm{pgl},2}\mathscr{L}_{\mathrm{pgl},2}(\sigma_y, \hat{\sigma}_y) + \cdots \tag{8.6}$$

Table 8.1: Parameters used for design exploration

Oracle Inputs

Property	Description
E	Material Elasticity
σ	Material Strength
υ	Poisson Ratio
ρ	Material Density
r_i	Inner Diameter
t	Thickness
l	Cylinder Length
σ_{hyd}	External Hydraulic Pressure
n	Number of Stiffeners
t_s	Thickness of Stiffener
h_s	Height of Stiffener
l_s	Location of Stiffener

Simulator Output

Property	Description	Notes:
f_r	feasibility ratio	σ_{max}/σ
$\sigma_{x,y,z}$	Maximum Directional Stresses	6 internal stress
$\sigma_{xy,yz,xz}$	Maximum Plane Stresses	for intermediate PGL variables

8.2.5 Implementation Details

For this study, we consider three capsule design, plain hull, hull with inner stiffeners, and hull with outer stiffeners. We assumed that the oracle is a simple design space explorer that uses Latin hypercube sampling approach (McKay et al., 2000). In total, 10,000 samples are obtained for each hull design from the design space with the lower and upper bounds tabulated in Table 8.2. The parameters related to stiffeners are also The material picked for the design is the structural steel with the following properties, $\sigma = 36\,ksi$, $E = 29,000\,ksi$, $\upsilon = 0.32$, $\rho = 0.284\,\text{lb/in}^3$. To reduce the simulation time and improve the computational efficiency, only 1/8 of the model is designed and simulated (see Figure 8.5). Symmetric boundary conditions are applied at the cross-section of the reduced capsule model to emulate full model behavior.

Table 8.2: Upper and lower bounds for parameters

Property	Lower Bound	Upper Bound	unit
r_i	7.5	20	in
t	0.125	1.125	in
l	20	50	in
σ_{hyd}	50	1500	psi
n	5	16	
t_s	0.125	1.125	in
h_s	0.125	1.125	in

After the data generation phase, the data is divided into training and testing data with a ratio of 4:1 and all of the features are standardized with respect to the training. Using the training dataset, a PGL-based surrogate model for each hull design is trained in PyTorch with the architectural layout described in Table 8.3. In parallel, a black-box model without the physics-gudied components is trained for the In addition, LRP rules defined in Table 8.3 are applied towards explainability.

Table 8.3: LRP layer rules

Layer Count	Layer Type	Notes	LRP Rule
1	Input		LRP-w^2
2	Linear		LRP-γ
3	ReLU		-
4	Linear		LRP-γ
5	ReLU		-
6	Linear	Intermediate	LRP-ε
7	ReLU		-
8	Linear		LRP-ε
9	ReLU		-
10	Linear		LRP-0
11	ReLU		-
12	Linear		-

8.3 Evaluation

8.4 A Brief Discussion on Generalization

To evaluate the generalization of the PGL model, a new set of data are drawn from outside of the explored design space. More specifically, we generated 100 samples from the space 10% above the upper bound and 100 samples from the space 10% below the lower bound tabulated in Table 8.2 using Latin hypercube sampling. The trained black-box and the PGL architectures are tested with the testing data from the design space as well as the data drawn outside the design space. We looked at a variety of metrics that evaluates the prediction performance of the design feasibility such as mean square error (*MSE* - see Eq 8.7), mean relative absolute error in percentage (*MAE* - see Eq 8.8), and maximum relative absolute error in percentage (AE_{max} - see Eq 8.9).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{8.7}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{8.8}$$

$$AE_{max} = \max(Y - \hat{Y}) \tag{8.9}$$

The Table 8.4 summarizes the performance of both black-box and PGL surrogate model for plain hull

design based on the aforementioned metrics. Here, *Original* and *Less Explored* designate the testing data within design space and the data outside the space, respectively. For all the metrics, PGL provides lower prediction error and better generalization over black-box. The results imply that PGL-based surrogate models could be used for computing simulation results cheaply outside of the explored design space to some extend.

Table 8.4: Generalization capability of black-box and PGL-based surrogate models

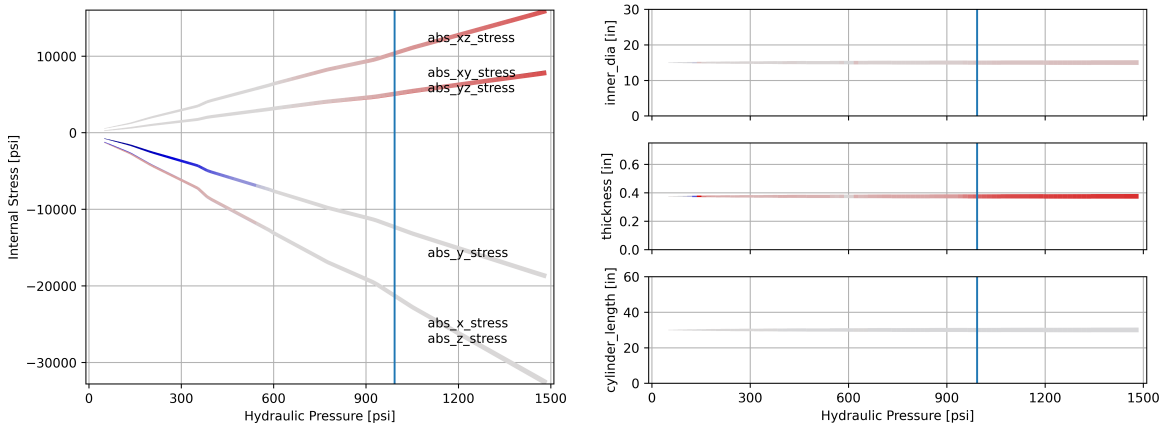
	Black-box			PGL		
Design Space	MSE	MAE	AE_{max}	MSE	MAE	AE_{max}
Original	0.0001395	2.21	9.43	**0.0000967**	**2.18**	**8.96**
Less Explored	0.0005468	8.66	16.12	**0.0000684**	**3.01**	**6.32**

8.5 Explainability of the Designs

As discussed before, three hull designs are considered. First, we look at the plain hull design. As an example, we pick a design from the design space with the following parameters: $r_i = 15.0$ in, $t = 0.375$ in, $l = 30.0$ in. Here, we examine the LRP scores for the intermediate physical values, i.e. the internal stresses (see Figure 8.6). To generate the LRP scores, we sweep the external hydraulic pressure from lower bound to upper bound (50-1500 psi). Furthermore, the external hydraulic pressure that causes the feasibility ratio to exceed 0.5 is marked with a blue vertical line.

The results for the given design parameters imply that for low levels of external hydraulic pressure, the directional stresses govern the feasibility. As the external hydraulic pressure increases, the effect of plane stresses (xz, xy, yz) become more dominant in the feasibility prediction. The relevancy scores for the design parameters indicate that the feasibility is driven mainly by the thickness parameter of the hull. The inner diameter has a secondary effect on the prediction, whereas the cylinder length does not affect the prediction drastically. The LRP results suggests that the designer should focus on calibrating thickness and inner diameter to fine-tune the feasibility for the given design parameters.

Figure 8.7 presents the relevancy scores for the hull designs with the stiffeners. The LRP results for intermediate stresses follow a pattern similar to plain hull design with more emphasis on the plane stresses at higher external hydraulic pressure. The relevancy scores for the design parameters imply that the hull thickness and the inner diamater are the most relevant to the feasibilty. Properties such as thickness, height, and number of stiffeners have a secondary relevance to the results. Compared to the plain hull design, the cylinder length has a weak effect on the feasibility for the designs with stiffeners.



(a) (b)

Figure 8.6: The progression of relevance for plain hull design: (a) intermediate physical parameters; (b) design parameters

8.6 Conclusion

This chapter presented a surrogate modeling method with physics-guided learning component. Additionally, this chapter discussed the explainability of surrogate models through layer-wise propagation. To evaluate the proposed method and the explainability of the surrogate model, we considered three hull designs, plain, hull body with inner stiffeners, and outer stiffeners. To reach to this aim, we first developed an automatic shape generation and simulation scheme to explore design space and develop a dataset. For each simulation, we collected geometric properties of the design as the input, and the feasibility, i.e. the ratio between maximum observed Von Mises stress and the material strength as the output. For each hull design, we generated a physics-guided surrogate model where the directional and plane stresses are selected as the intermediate physics-related variables. The results demonstrated that the PGL-based surrogate model promises a better generalization for less explored design space compared to a black-box approach. As for interpretability, an analysis of LRP scores points out that the proposed method allows the designer to explain which design parameters are most dominant in fine-tuning of the feasibility.

8.7 Future Work

As a future research, a single surrogate model for the three hull designs should be trained to explain the relevancy of all design parameters in a more complete scheme. In addition, the effect of intermediate parameters such as displacements and strains should be introduced into the PGL and the relevancy of such parameters on the output should be investigated.

While LRP provides an explanation on how physical parameters affect the prediction *qualitatively*, its

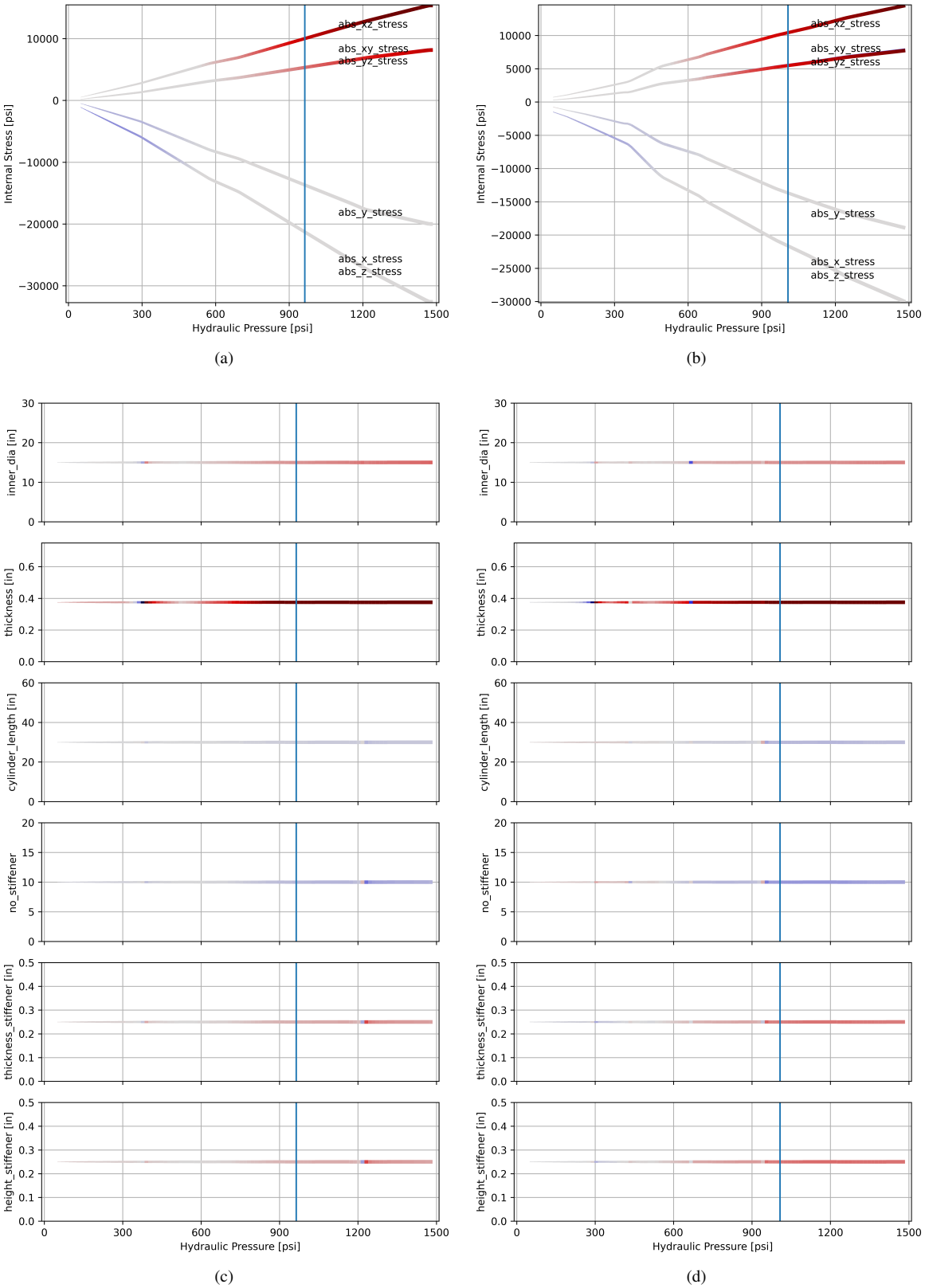


Figure 8.7: The progression of relevance for hull design with stiffeners: (a) intermediate physical parameters - inside stiffener; (b) design parameters - inside stiffener; (c) intermediate physical parameters - outside stiffener; (d) design parameters - outside stiffener

ability in explaining sensitivity of individual inputs on the output *quantitatively* is limited. This limitation may hinder the symbiotic design processes involving multiple domain expertise where human feedback towards design optimization is crucial. Future work should look into methods where LRP scores are attributed to sensitivity of input parameters for more informed design decisions.

CHAPTER 9

Conclusion

In the last few decades, with the introduction of machine learning, the research on design and monitoring of civil structures and mechanical systems has evolved into a new era. A majority of such ML-based engineering applications adopts a black-box approach with the assumption that the training and testing data share the same probabilistic distributions. However, this expectation is often unrealistic for many cases, since the access to the complete labeled training data can be very limited, especially when the system is newly deployed. One can employ computer simulations to generate simulated training data to compensate for the absence of experimental data. However, there is no guarantee that the simulations will yield high-fidelity results. Furthermore, structural and mechanical systems are dynamic by nature and usually experience change during their life time. As result of this, the divergence between training and testing data is imminent and may lead to prediction errors and compromise the safety of the system.

This dissertation proposes a set of methods to reduce the probabilistic divergence between training and testing data and to improve the overall generalization of the machine learning algorithms. More specifically, this work seeks to apply the domain adaptation methods into the damage detection and localization problems to transfer the knowledge from a well explored source domain to the unlabeled target domain. Another approach considered for improving generalization of damage detection algorithms susceptible to source-target divergence is the integration of physics-based domain knowledge into the learning, which is known as physics-guided learning.

Another issue with ML-based applications is that black-box models are by nature non-transparent to the end user. Thus, it is often hard to interpret the reasoning of the algorithm. This work proposes an interpretable physics-guided learning approach that utilizes domain-specific physics knowledge to create *glass-box models* and to extend the explainability through intermediate physics-related parameters. Lastly, this work looks into the explainable neural network based surrogate models with physics guided learning components. The proposed method allows the designers to have a better understanding and explainability over the design process.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., and Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388:154–170.

Alaiz-Rodríguez, R. and Japkowicz, N. (2008). Assessing the impact of changing environments on classifier performance. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 13–24. Springer.

Alam, K. M., Siddique, N., Adeli, H., et al. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, 32(12):8675–8690.

Amarnath, M., Sugumaran, V., and Kumar, H. (2013). Exploiting sound signals for fault diagnosis of bearings using decision tree. *Measurement*, 46(3):1250–1256.

American Society of Civil Engineers (2013). Failure to act. https://www.asce.org/uploadedFiles/Issues_and_Advocacy/Our_Initiatives/Infrastructure/Content_Pieces/failure-to-act-economic-impact-summary-report.pdf. Accessed: 2020-05-15.

American Society of Civil Engineers (2017). 2017 infrastructure report card. https://www.infrastructurereportcard.org/wp-content/uploads/2019/02/Full-2017-Report-Card-FINAL.pdf. Accessed: 2020-05-15.

Amezquita-Sanchez, J. P. and Adeli, H. (2015). A new music-empirical wavelet transform methodology for time–frequency analysis of noisy nonlinear and non-stationary signals. *Digital Signal Processing*, 45:55–68.

Amezquita-Sanchez, J. P., Park, H. S., and Adeli, H. (2017). A novel methodology for modal parameters identification of large smart structures using music, empirical wavelet transform, and hilbert transform. *Engineering Structures*, 147:148–159.

ANSYS (2022). *Student Version, 2022 R2*. Ansys, Inc., Canonsburg, Pennsylvania.

Antwarg, L., Miller, R. M., Shapira, B., and Rokach, L. (2019). Explaining anomalies detected by autoencoders using shap. *arXiv preprint arXiv:1903.02407*.

Audet, C., Denni, J., Moore, D., Booker, A., and Frank, P. (2000). A surrogate-model-based method for constrained optimization. In *8th symposium on multidisciplinary analysis and optimization*, page 4891.

Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., and Inman, D. J. (2021). A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications. *Mechanical systems and signal processing*, 147:107077.

Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., and Inman, D. J. (2018). Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks. *Journal of Sound and Vibration*, 424:158–172.

Avci, O., Abdeljaber, O., Kiranyaz, S., and Inman, D. (2017). Structural damage detection in real time: implementation of 1d convolutional neural networks for shm applications. In *Structural Health Monitoring & Damage Detection, Volume 7*, pages 49–54. Springer.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Bajer, L. and Holeňa, M. (2010). Surrogate model for continuous and discrete genetic optimization based on rbf networks. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 251–258. Springer.

Bakhary, N., Hao, H., and Deeks, A. J. (2007). Damage detection using artificial neural network with consideration of uncertainties. *Engineering Structures*, 29(11):2806–2815.

Balageas, D., Fritzen, C.-P., and Güemes, A. (2010). *Structural health monitoring*, volume 90. John Wiley & Sons.

Bárkányi, Á., Chován, T., Németh, S., and Abonyi, J. (2021). Modelling for digital twins—potential role of surrogate models. *Processes*, 9(3):476.

Belle, V. and Papantonis, I. (2021). Principles and practice of explainable machine learning. *Frontiers in big Data*, page 39.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*, 79(1-2):151–175.

Benaim, S. and Wolf, L. (2017). One-sided unsupervised domain mapping. In *Advances in neural information processing systems*, pages 752–762.

Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J. M., and Eckersley, P. (2020). Explainable machine learning in deployment. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 648–657.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J. (2006). Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57.

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., and Krishnan, D. (2017). Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731.

Bouvier, V., Very, P., Hudelot, C., and Chastagnol, C. (2019). Hidden covariate shift: A minimal assumption for domain adaptation. *arXiv preprint arXiv:1907.12299*.

Brincker, R., Zhang, L., and Andersen, P. (2000). Modal identification from ambient responses using frequency domain decomposition. In *Proc. of the 18*'International Modal Analysis Conference (IMAC), San Antonio, Texas*.

Brownjohn, J. M. W. (2003). Ambient vibration studies for system identification of tall buildings. *Earthquake engineering & structural dynamics*, 32(1):71–95.

Bruzzone, A. G., Merani, D., Massei, M., Tremori, A., Bartolucci, C., and Ferrando, A. (2013). Modeling cyber warfare in heterogeneous networks for protection of infrastructures and operations. *Proceedings of I3M2013, Athens, Greece*.

Busby, D., Farmer, C. L., and Iske, A. (2007). Hierarchical nonlinear approximation for experimental design and statistical data fitting. *SIAM Journal on Scientific Computing*, 29(1):49–69.

Caicedo, J. M. (2011). Practical guidelines for the natural excitation technique (next) and the eigensystem realization algorithm (era) for modal identification using ambient vibration. *Experimental Techniques*, 35(4):52–58.

Caicedo, J. M., Dyke, S. J., and Johnson, E. A. (2004). Natural excitation technique and eigensystem realization algorithm for phase i of the iasc-asce benchmark problem: Simulated data. *Journal of Engineering Mechanics*, 130(1):49–60.

Caicedo, J. M., Marulanda, J., Thomson, P., and Dyke, S. J. (2001). Monitoring of bridges to detect changes in structural health. In *American Control Conference, 2001. Proceedings of the 2001*, volume 1, pages 453–458. IEEE.

Catbas, N., Gokce, H. B., and Frangopol, D. M. (2013). Predictive analysis by incorporating uncertainty through a family of models calibrated with structural health-monitoring data. *Journal of Engineering Mechanics*, 139(6):712–723.

Chalapathy, R. and Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.

Chang, W.-G., You, T., Seo, S., Kwak, S., and Han, B. (2019). Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7354–7362.

Chattopadhyay, A. and Lam, K.-Y. (2017). Security of autonomous vehicle as a cyber-physical system. In *2017 7th International Symposium on Embedded Computing and System Design (ISED)*, pages 1–6. IEEE.

Chen, B., Yang, Z., Huang, S., Du, X., Cui, Z., Bhimani, J., Xie, X., and Mi, N. (2017). Cyber-physical system enabled nearby traffic flow modelling for autonomous vehicles. In *2017 IEEE 36th international performance computing and communications conference (IPCCC)*, pages 1–6. IEEE.

Chen, S., Ju, M.-S., and Tsuei, Y. (1996). Estimation of mass, stiffness and damping matrices from frequency response functions. *Journal of Vibration, Acoustics, Stress, and Reliability in Design*, 118(1):78–82.

Chen, X., Xu, X., Liu, X., Pan, S., He, J., Noh, H. Y., Zhang, L., and Zhang, P. (2018). Pga: Physics guided and adaptive approach for mobile fine-grained air pollution estimation. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 1321–1330.

Chen, X.-y. and Zhan, Y.-y. (2008). Multi-scale anomaly detection algorithm based on infrequent pattern of time series. *Journal of Computational and Applied Mathematics*, 214(1):227–237.

Chen, Z., Li, C., and Sanchez, R.-V. (2015). Gearbox fault identification and classification with convolutional neural networks. *Shock and Vibration*, 2015.

Chollet, F. (2015). Keras. https://keras.io.

Cobb, A., Roy, A., Elenius, D., and Jha, S. (2022). Trinity ai co-designer for hierarchical oracle-guided design of cyber-physical systems. In *2022 IEEE Workshop on Design Automation for CPS and IoT (DESTION)*, pages 42–44. IEEE.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Craig Jr, R. R. and Kurdila, A. J. (2006). *Fundamentals of structural dynamics*. John Wiley & Sons.

Crombecq, K. (2011). *Surrogate modeling of computer experiments with sequential experimental design*. PhD thesis, Ghent University.

Crombecq, K., Couckuyt, I., Gorissen, D., and Dhaene, T. (2009a). Space-filling sequential design strategies for adaptive surrogate modelling. In *The first international conference on soft computing technology in civil, structural and environmental engineering*, volume 38.

Crombecq, K., De Tommasi, L., Gorissen, D., and Dhaene, T. (2009b). A novel sequential design strategy for global surrogate modeling. In *Proceedings of the 2009 winter simulation conference (WSC)*, pages 731–742. IEEE.

Crombecq, K., Gorissen, D., Deschrijver, D., and Dhaene, T. (2011a). A novel hybrid sequential design strategy for global surrogate modeling of computer experiments. *SIAM Journal on Scientific Computing*, 33(4):1948–1974.

Crombecq, K., Laermans, E., and Dhaene, T. (2011b). Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. *European Journal of Operational Research*, 214(3):683–696.

Dackermann, U., Li, J., and Samali, B. (2013). Identification of member connectivity and mass changes on a two-storey framed structure using frequency response functions and artificial neural networks. *Journal of Sound and Vibration*, 332(16):3636–3653.

Davis, S. E., Cremaschi, S., and Eden, M. R. (2018). Efficient surrogate model development: Impact of sample size and underlying model dimensions. In *Computer Aided Chemical Engineering*, volume 44, pages 979–984. Elsevier.

Daw, A., Karpatne, A., Watkins, W. D., Read, J. S., and Kumar, V. (2017). Physics-guided neural networks (pgnn): An application in lake temperature modeling. In *Knowledge-Guided Machine Learning*, pages 353–372. Chapman and Hall/CRC.

Daw, A., Thomas, R. Q., Carey, C. C., Read, J. S., Appling, A. P., and Karpatne, A. (2020). Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling. In *Proceedings of the 2020 siam international conference on data mining*, pages 532–540. SIAM.

Deraemaeker, A. and Worden, K. (2018). A comparison of linear approaches to filter out environmental effects in structural health monitoring. *Mechanical systems and signal processing*, 105:1–15.

Dervilis, N., Barthorpe, R. J., Antoniadou, I., Staszewski, W. J., and Worden, K. (2012a). Damage detection in carbon composite material typical of wind turbine blades using auto-associative neural networks. In *Health Monitoring of Structural and Biological Systems 2012*, volume 8348, page 834806. International Society for Optics and Photonics.

Dervilis, N., Choi, M., Antoniadou, I., Farinholt, K. M., Taylor, S. G., Barthorpe, R. J., Park, G., Worden, K., and Farrar, C. R. (2012b). Novelty detection applied to vibration data from a cx-100 wind turbine blade under fatigue loading. *Journal of Physics: Conference Series*, 382(1):012047.

Devin, A. and Fanning, P. (2012). Impact of nonstructural components on modal response and structural damping. In *Topics on the Dynamics of Civil Structures, Volume 1*, pages 415–421. Springer.

Doebling, S. W., Farrar, C. R., Prime, M. B., and Shevitz, D. W. (1996). Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review. *Los Alamos National Laboratory, Los Alamos, NM, Report No. LA-13070*.

Dong, S., Luo, T., Zhong, L., Chen, L., and Xu, X. (2017). Fault diagnosis of bearing based on the kernel principal component analysis and optimized k-nearest neighbour model. *Journal of Low Frequency Noise, Vibration and Active Control*, 36(4):354–365.

Eason, J. and Cremaschi, S. (2014). Adaptive sequential sampling for surrogate model generation with artificial neural networks. *Computers & Chemical Engineering*, 68:220–232.

Emerson, M. and Sztipanovits, J. (2006). Techniques for metamodel composition. In *OOPSLA–6th Workshop on Domain Specific Modeling*, pages 123–139.

Farhang-Mehr, A. and Azarm, S. (2005). Bayesian meta-modelling of engineering design simulations: a sequential approach with adaptation to irregularities in the response behaviour. *International Journal for Numerical Methods in Engineering*, 62(15):2104–2126.

Farrar, C. R., Doebling, S. W., Cornwell, P. J., and Straser, E. G. (1996). Variability of modal parameters measured on the alamosa canyon bridge. Technical report, Los Alamos National Lab., NM (United States).

Farrar, C. R. and Worden, K. (2007). An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):303–315.

Farrar, C. R. and Worden, K. (2012). *Structural health monitoring: a machine learning perspective*. John Wiley & Sons.

Figueiredo, E., Moldovan, I., Santos, A., Campos, P., and Costa, J. C. (2019). Finite element–based machine-learning approach to detect damage in bridges under operational and environmental variations. *Journal of Bridge Engineering*, 24(7):04019061.

Figueiredo, E., Park, G., Figueiras, J., Farrar, C., and Worden, K. (2009). Structural health monitoring algorithm comparisons using standard data sets. *Los Alamos National Laboratory, Los Alamos, NM, Report No. LA-14393*.

Forssell, U. and Lindskog, P. (1997). Combining semi-physical and neural network modeling: An example ofits usefulness. *IFAC Proceedings Volumes*, 30(11):767–770.

Fritzen, C.-P. (1986). Identification of Mass, Damping, and Stiffness Matrices of Mechanical Systems. *Journal of Vibration, Acoustics, Stress, and Reliability in Design*, 108(1):9–16.

Frye, C., Rowat, C., and Feige, I. (2020). Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems*, 33:1229–1239.

Fuentes, R., Dwyer-Joyce, R., Marshall, M., Wheals, J., and Cross, E. (2020). Detection of sub-surface damage in wind turbine bearings using acoustic emissions and probabilistic modelling. *Renewable Energy*, 147:776–797.

Fukunaga, K. and Koontz, W. L. G. (1970). Application of the karhunen-loeve expansion to feature selection and ordering. *IEEE Transactions on Computers*, C-19(4):311–318.

Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Garbo, A. and German, B. (2017). Adaptive sampling with adaptive surrogate model selection for computer experiment applications. In *18th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 4430.

Gardner, P., Bull, L., Dervilis, N., and Worden, K. (2021). Overcoming the problem of repair in structural health monitoring: Metric-informed transfer learning. *Journal of Sound and Vibration*, 510:116245.

Gardner, P., Liu, X., and Worden, K. (2020). On the application of domain adaptation in structural health monitoring. *Mechanical Systems and Signal Processing*, 138:106550.

Ghanem, R. and Shinozuka, M. (1995). Structural-system identification. i: Theory. *Journal of Engineering Mechanics*, 121(2):255–264.

Giraldo, D., Yoshida, O., Dyke, S. J., and Giacosa, L. (2004). Control-oriented system identification using era. *Structural Control and Health Monitoring*, 11(4):311–326.

Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE.

Gonzalez, I. (2014). *Application of monitoring to dynamic characterization and damage detection in bridges*. PhD thesis, KTH, Structural Engineering and Bridges. QC 20140910.

Goodfellow, I., Bengio, Y., and Courville, A. (2016a). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goodfellow, I., Bengio, Y., and Courville, A. (2016b). *Deep learning*. MIT press.

Gramacy, R. B. and Lee, H. K. (2018). Adaptive design of supercomputer experiments.

Gres, S., Ulriksen, M. D., Döhler, M., Johansen, R. J., Andersen, P., Damkilde, L., and Nielsen, S. A. (2017). Statistical methods for damage detection applied to civil structures. *Procedia engineering*, 199:1919–1924.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2006). A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19:513–520.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5.

Gu, J., Gul, M., and Wu, X. (2017). Damage detection under varying temperature using artificial neural networks. *Structural Control and Health Monitoring*, 24(11):e1998. e1998 STC-15-0016.R3.

Gulgec, N. S., Takáč, M., and Pakzad, S. N. (2017). Structural damage detection using convolutional neural networks. In *Model Validation and Uncertainty Quantification, Volume 3*, pages 331–337. Springer.

Gulgec, N. S., Takáč, M., and Pakzad, S. N. (2019). Convolutional neural network approach for robust structural damage detection and localization. *Journal of computing in civil engineering*, 33(3):04019005.

Guo, Q., Feng, L., Zhang, R., and Yin, H. (2020). Study of damage identification for bridges based on deep belief network. *Advances in Structural Engineering*, 23(8):1562–1572.

Gutmann, H.-M. (2001). A radial basis function method for global optimization. *Journal of global optimization*, 19(3):201–227.

Hakim, S. and Razak, H. A. (2014). Modal parameters based structural damage detection using artificial neural networks-a review. *Smart Structures and Systems*, 14(2):159–189.

Hallaq, B., Somer, T., Osula, A.-M., Ngo, K., and Mitchener-Nissen, T. (2017). Artificial intelligence within the military domain and cyber warfare. In *Eur. Conf. Inf. Warf. Secur. ECCWS*, pages 153–157.

Halton, J. H. (1964). Algorithm 247: Radical-inverse quasi-random point sequence. *Communications of the ACM*, 7(12):701–702.

Han, X., Xiang, H., Li, Y., and Wang, Y. (2019). Predictions of vertical train-bridge response using artificial neural network-based surrogate model. *Advances in Structural Engineering*, 22(12):2712–2723.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, Y.-L., Wang, R., Kwong, S., and Wang, X.-Z. (2014). Bayesian classifiers based on probability density estimation and their applications to simultaneous fault diagnosis. *Information Sciences*, 259:252–268.

Hernandez-Garcia, M. R., Masri, S. F., Ghanem, R., Figueiredo, E., and Farrar, C. R. (2010). An experimental investigation of change detection in uncertain chain-like systems. *Journal of Sound and Vibration*, 329(12):2395–2409.

Hickernell, F. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of computation*, 67(221):299–322.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR.

Holzinger, A. (2018). From machine learning to explainable ai. In *2018 world symposium on digital intelligence for systems and machines (DISA)*, pages 55–66. IEEE.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. (2006). Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601–608.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. J. (2007). Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608.

Huang, Q., Yamada, M., Tian, Y., Singh, D., and Chang, Y. (2022). Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*.

Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE transactions on information theory*, 14(1):55–63.

Husslage, B., Van Dam, E., and Den Hertog, D. (2005). Nested maximin latin hypercube designs in two dimensions.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

Islam, M. M. and Kim, J.-M. (2019). Automated bearing fault diagnosis scheme using 2d representation of wavelet packet transform and deep convolutional neural network. *Computers in Industry*, 106:142–153.

Jaishi, B. and Ren, W.-X. (2006). Damage detection by finite element model updating using modal flexibility residual. *Journal of sound and vibration*, 290(1-2):369–387.

James, G. H., Carne, T. G., and Lauffer, J. P. (1993). The natural excitation technique (next) for modal parameter extraction from operating wind turbines. *Sandia National Laboratories, Albuquerque, NM, Report No. SAND92-1666*.

James, G. H., Carne, T. G., and Lauffer, J. P. (1995). The natural excitation technique (next) for modal parameter extraction from operating structures. *Modal Analysis-the International Journal of Analytical and Experimental Modal Analysis*, 10(4):260.

Jia, F., Lei, Y., Lin, J., Zhou, X., and Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, 72:303–315.

Jia, X., Karpatne, A., Willard, J., Steinbach, M., Read, J., Hanson, P. C., Dugan, H. A., and Kumar, V. (2018). Physics guided recurrent neural networks for modeling dynamical systems: Application to monitoring water temperature and quality in lakes. *arXiv preprint arXiv:1810.02880*.

Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M., and Kumar, V. (2019). Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 558–566. SIAM.

Jiang, X. and Adeli, H. (2007). Pseudospectra, music, and dynamic wavelet neural network for damage detection of highrise buildings. *International Journal for Numerical Methods in Engineering*, 71(5):606–629.

Jin, R., Chen, W., and Sudjianto, A. (2002). On sequential sampling for global metamodeling in engineering design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 36223, pages 539–548.

Jing, L., Zhao, M., Li, P., and Xu, X. (2017). A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. *Measurement*, 111:1–10.

Jones, D., Snider, C., Nassehi, A., Yon, J., and Hicks, B. (2020). Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29:36–52.

Juang, J. and Pappa, R. S. (1985). An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of guidance, control, and dynamics*, 8(5):620–627.

Kani, J. N. and Elsheikh, A. H. (2017). Dr-rnn: A deep residual recurrent neural network for model reduction. *arXiv preprint arXiv:1709.00939*.

Karpatne, A., Watkins, W., Read, J., and Kumar, V. (2017). Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*.

Kaszynski, A. (2020). pyansys: Python Interface to MAPDL and Associated Binary and ASCII Files.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.

Kim, J.-T., Ryu, Y.-S., Cho, H.-M., and Stubbs, N. (2003). Damage identification in beam-type structures: frequency-based method vs mode-shape-based method. *Engineering structures*, 25(1):57–67.

Kim, S. J., Christenson, R., Phillips, B., and Spencer Jr, B. (2012). Geographically distributed real-time hybrid simulation of mr dampers for seismic hazard mitigation. In *Proceedings of the 20th Analysis and Computation Specialty Conference, Chicago, IL, USA*, pages 29–31.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, D. J. (2019). 1d convolutional neural networks and applications: A survey. *arXiv preprint arXiv:1905.03554*.

Kleijnen, J. P. and Van Beers, W. C. (2004). Application-driven sequential designs for simulation experiments: Kriging metamodelling. *Journal of the operational research society*, 55(8):876–883.

Kodirov, E., Xiang, T., Fu, Z., and Gong, S. (2015). Unsupervised domain adaptation for zero-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Kopsaftopoulos, F. P. and Fassois, S. D. (2013). A functional model based statistical time series method for vibration based damage detection, localization, and magnitude estimation. *Mechanical Systems and Signal Processing*, 39(1-2):143–161.

Koziel, S. and Leifsson, L. (2013). Surrogate-based aerodynamic shape optimization by variable-resolution models. *AIAA journal*, 51(1):94–106.

Kramer, M. A. (1991a). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243.

Kramer, M. A. (1991b). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243.

Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise.

Lal, A. and Datta, B. (2018). Development and implementation of support vector machine regression surrogate models for predicting groundwater pumping-induced saltwater intrusion into coastal aquifers. *Water Resources Management*, 32(7):2405–2419.

LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer.

Lee, C.-Y., Batra, T., Baig, M. H., and Ulbricht, D. (2019a). Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10285–10295.

Lee, J. J., Lee, J. W., Yi, J. H., Yun, C. B., and Jung, H. Y. (2005). Neural networks-based damage detection for bridges considering errors in baseline finite element models. *Journal of Sound and Vibration*, 280(3-5):555–578.

Lee, S., Kim, D., Kim, N., and Jeong, S.-G. (2019b). Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 91–100.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., and Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138:106587.

Lei, Y. and Zuo, M. J. (2009). Gear crack level identification based on weighted k nearest neighbor classification algorithm. *Mechanical Systems and Signal Processing*, 23(5):1535–1547.

Lei, Y., Zuo, M. J., He, Z., and Zi, Y. (2010). A multidimensional hybrid intelligent method for gear fault diagnosis. *Expert Systems with Applications*, 37(2):1419–1430.

Li, J. (2014). *Structural health monitoring of an In-service highway bridge with uncertainties*. PhD thesis, University of Connecticut.

Li, J., Li, X., He, D., and Qu, Y. (2020). A domain adaptation model for early gear pitting fault diagnosis based on deep transfer learning network. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 234(1):168–182.

Li, X., Ozdagli, A. I., Dyke, S. J., Lu, X., and Christenson, R. (2017). Development and verification of distributed real-time hybrid simulation methods. *Journal of Computing in Civil Engineering*, 31(4):04017014.

Li, X., Zhang, W., and Ding, Q. (2018). Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks. *IEEE Transactions on Industrial Electronics*, 66(7):5525–5534.

Li, X., Zhang, W., Ding, Q., and Sun, J.-Q. (2019). Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal processing*, 157:180–197.

Li, Y., Wang, N., Shi, J., Liu, J., and Hou, X. (2016). Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*.

Liang, Y., Wu, D., Liu, G., Li, Y., Gao, C., Ma, Z. J., and Wu, W. (2016). Big data-enabled multiscale serviceability analysis for aging bridges. *Digital Communications and Networks*, 2(3):97–107.

Lin, Y. (2004). *An efficient robust concept exploration method and sequential exploratory experimental design*. PhD thesis, Georgia Institute of Technology.

Lin, Y.-z., Nie, Z.-h., and Ma, H.-w. (2017). Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, 32(12):1025–1046.

Lin, Y.-z., Nie, Z.-h., and Ma, H.-w. (2022). Dynamics-based cross-domain structural damage detection through deep transfer learning. *Computer-Aided Civil and Infrastructure Engineering*, 37(1):24–54.

Lin, Z., Pan, H., Wang, X., and Li, M. (2018). Data-driven structural diagnosis and conditional assessment: From shallow to deep learning. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018*, volume 10598, page 1059814. International Society for Optics and Photonics.

Liu, B., Zhang, Q., and Gielen, G. G. (2013). A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Transactions on Evolutionary Computation*, 18(2):180–192.

Liu, C. and DeWolf, J. T. (2007). Effect of temperature on modal variability of a curved concrete bridge under ambient loads. *Journal of Structural Engineering*, 133(12):1742–1751.

Liu, D. and Wang, Y. (2019). Multi-fidelity physics-constrained neural network and its application in materials modeling. *Journal of Mechanical Design*, 141(12).

Liu, H., Li, L., and Ma, J. (2016a). Rolling bearing fault diagnosis based on stft-deep learning and sound signals. *Shock and Vibration*, 2016.

Liu, H., Xu, S., Ma, Y., Chen, X., and Wang, X. (2016b). An adaptive bayesian sequential sampling approach for global metamodeling. *Journal of Mechanical Design*, 138(1).

Loeppky, J. L., Sacks, J., and Welch, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51(4):366–376.

Long, M., Cao, Y., Cao, Z., Wang, J., and Jordan, M. I. (2018). Transferable representation learning with deep adaptation networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):3071–3085.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. (2017). Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*.

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. (2013). Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207.

Lu, C., Wang, Z.-Y., Qin, W.-L., and Ma, J. (2017). Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Processing*, 130:377–388.

Lu, W., Liang, B., Cheng, Y., Meng, D., Yang, J., and Zhang, T. (2016). Deep model based domain adaptation for fault diagnosis. *IEEE Transactions on Industrial Electronics*, 64(3):2296–2305.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Ma, S., Chu, F., and Han, Q. (2019). Deep residual learning with demodulated time-frequency features for fault diagnosis of planetary gearbox under nonstationary running conditions. *Mechanical Systems and Signal Processing*, 127:190–201.

Ma, X., Lin, Y., Nie, Z., and Ma, H. (2020). Structural damage identification based on unsupervised feature-extraction via variational auto-encoder. *Measurement*, 160:107811.

Marec, A., Thomas, J.-H., and El Guerjouma, R. (2008). Damage characterization of polymer-based composite materials: Multivariable analysis and wavelet transform for clustering acoustic emission data. *Mechanical systems and signal processing*, 22(6):1441–1464.

Mariniello, G., Pastore, T., Menna, C., Festa, P., and Asprone, D. (2020). Structural damage detection and localization using decision tree ensemble and vibration data. *Computer-Aided Civil and Infrastructure Engineering*.

Mason, J. D., Ayorinde, E. T., Mascarenas, D. D., and Moreu, F. (2016). Tap testing hammer using unmanned aerial systems (uass). Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Matarazzo, T. J., Shahidi, S. G., Chang, M., and Pakzad, S. N. (2015). Are today's shm procedures suitable for tomorrow's bigdata? In *Structural Health Monitoring and Damage Detection, Volume 7*, pages 59–65. Springer.

MATLAB (2018). *version 9.5.0 (R2018b)*. The MathWorks Inc., Natick, Massachusetts.

McKay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.

McKenna, F., Scott, M. H., and Fenves, G. L. (2010). Nonlinear finite-element analysis software architecture using object composition. *Journal of Computing in Civil Engineering*, 24(1):95–107.

Mehrjoo, M., Khaji, N., Moharrami, H., and Bahreininejad, A. (2008). Damage detection of truss bridge joints using artificial neural networks. *Expert Systems with Applications*, 35(3):1122–1131.

Mirzaee, A., Abbasnia, R., and Shayanfar, M. (2015). A comparative study on sensitivity-based damage detection methods in bridges. *Shock and Vibration*, 2015.

Mises, R. v. (1913). Mechanik der festen körper im plastisch-deformablen zustand. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1913:582–592.

Mohan, A. T. and Gaitonde, D. V. (2018). A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks. *arXiv preprint arXiv:1804.09269*.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., and Müller, K.-R. (2019). Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209.

Mora, L., Wu, X., and Panori, A. (2020). Mind the gap: Developments in autonomous driving research and the sustainability challenge. *Journal of cleaner production*, 275:124087.

Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530.

Moughty, J. J. and Casas, J. R. (2017). A state of the art review of modal-based damage detection in bridges: Development, challenges, and solutions. *Applied Sciences*, 7(5):510.

Muralidhar, N., Bu, J., Cao, Z., He, L., Ramakrishnan, N., Tafti, D., and Karpatne, A. (2019). Physics-guided design and learning of neural networks for predicting drag force on particle suspensions in moving fluids. *arXiv preprint arXiv:1911.04240*.

Muralidharan, V. and Sugumaran, V. (2012). A comparative study of naïve bayes classifier and bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis. *Applied Soft Computing*, 12(8):2023–2029.

Ni, Y., Zhou, H., Chan, K., and Ko, J. (2008). Modal flexibility analysis of cable-stayed ting kau bridge for damage identification. *Computer-Aided Civil and Infrastructure Engineering*, 23(3):223–236.

Ni, Y. Q., Hua, X. G., Fan, K. Q., and Ko, J. M. (2005). Correlating modal properties with temperature using long-term monitoring data and support vector machine technique. *Engineering Structures*, 27(12):1762–1773. SEMC 2004 Structural Health Monitoring, Damage Detection and Long-Term Performance.

Nick, W., Asamene, K., Bullock, G., Esterline, A., and Sundaresan, M. (2015). A study of machine learning techniques for detecting and classifying structural damage. *International Journal of Machine Learning and Computing*, 5(4):313.

Oh, B. K., Kim, K. J., Kim, Y., Park, H. S., and Adeli, H. (2017). Evolutionary learning based sustainable strain sensing model for structural health monitoring of high-rise buildings. *Applied Soft Computing*, 58:576–585.

Osio, I. G. and Amon, C. H. (1996). An engineering design methodology with multistage bayesian surrogates and optimal sampling. *Research in Engineering Design*, 8(4):189–206.

Owen, A. B. (1992). Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, pages 439–452.

Ozdagli, A. and Koutsoukos, X. (2020). Domain adaptation for structural health monitoring. In *Annual Conference of the PHM Society*, volume 12, pages 9–9.

Ozdagli, A. and Koutsoukos, X. (2021a). Domain adaptation for structural fault detection under model uncertainty. *International Journal of Prognostics and Health Management*, 12(2).

Ozdagli, A. I. (2015). *Distributed real-time hybrid simulation: Modeling, development and experimental validation*. PhD thesis, Purdue University.

Ozdagli, A. I. and Koutsoukos, X. (2019). Machine learning based novelty detection using modal analysis. *Computer-Aided Civil and Infrastructure Engineering*, 34(12):1119–1140.

Ozdagli, A. I. and Koutsoukos, X. (2021b). Model-based damage detection through physics guided learning. In *Annual Conference of the PHM Society*, volume 13.

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2010). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Pandey, A., Biswas, M., and Samman, M. (1991). Damage detection from changes in curvature mode shapes. *Journal of sound and vibration*, 145(2):321–332.

Paolucci, R., Gatti, F., Infantino, M., Smerzini, C., Özcebe, A. G., and Stupazzini, M. (2018). Broadband ground motions from 3d physics-based numerical simulations using artificial neural networksbroadband ground motions from 3d pbss using anns. *Bulletin of the Seismological Society of America*, 108(3A):1272–1286.

Papadopoulos, V., Soimiris, G., Giovanis, D., and Papadrakakis, M. (2018). A neural network-based surrogate model for carbon nanotubes with geometric nonlinearities. *Computer Methods in Applied Mechanics and Engineering*, 328:411–430.

Parish, E. J. and Duraisamy, K. (2016). A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774.

Park, J.-H., Kim, J.-T., Hong, D.-S., Ho, D.-D., and Yi, J.-H. (2009). Sequential damage detection approaches for beams using time-modal features and artificial neural networks. *Journal of Sound and Vibration*, 323(1-2):451–474.

Patel, V. M., Gopalan, R., Li, R., and Chellappa, R. (2015). Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peeters, B. and Roeck, G. D. (1999). Reference-based stochastic subspace identification for output-only modal analysis. *Mechanical Systems and Signal Processing*, 13(6):855–878.

Pei, Z., Cao, Z., Long, M., and Wang, J. (2018). Multi-adversarial domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*, volume 32.

Peng, K.-C., Wu, Z., and Ernst, J. (2018). Zero-shot deep domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J. P., and Adeli, H. (2020). Fema: A finite element machine for fast learning. *Neural Computing and Applications*, 32(10):6393–6404.

Perez-Ramirez, C. A., Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., Adeli, H., Dominguez-Gonzalez, A., and Romero-Troncoso, R. J. (2019). Recurrent neural network model with bayesian training and mutual information for response prediction of large buildings. *Engineering Structures*, 178:603–615.

Phm Society (2009). Phm data challenge 2009. http://www.phmsociety.org/competition/09. Accessed: 2009-09-28.

Qian, P. Z. (2009). Nested latin hypercube designs. *Biometrika*, 96(4):957–970.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K. (2005). Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28.

Quiñonero-Candela, J., Sugiyama, M., Lawrence, N. D., and Schwaighofer, A. (2009). *Dataset shift in machine learning*. Mit Press.

Rafiei, M. H. and Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE transactions on neural networks and learning systems*, 28(12):3074–3083.

Ratto, M. and Pagano, A. (2010). Using recursive algorithms for the efficient identification of smoothing spline anova models. *AStA Advances in Statistical Analysis*, 94(4):367–388.

Redko, I., Morvant, E., Habrard, A., Sebban, M., and Bennani, Y. (2020). A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv e-prints*, pages arXiv–2004.

Rennen, G., Husslage, B., Van Dam, E. R., and Den Hertog, D. (2010). Nested maximin latin hypercube designs. *Structural and Multidisciplinary Optimization*, 41(3):371–395.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Roscher, R., Bohn, B., Duarte, M. F., and Garcke, J. (2020). Explainable machine learning for scientific insights and discoveries. *Ieee Access*, 8:42200–42216.

Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. (2008). Incremental learning for robust visual tracking. *International journal of computer vision*, 77(1-3):125–141.

Rossum, G. (1995). Python tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Rytter, A. (1993). *Vibrational based inspection of civil engineering structures*. PhD thesis, Dept. of Building Technology and Structural Engineering, Aalborg University.

S., H. (2007). Effects of environmental and operational variability on structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):539–560.

Sadoughi, M. and Hu, C. (2019). Physics-based convolutional neural network for fault diagnosis of rolling element bearings. *IEEE Sensors Journal*, 19(11):4181–4192.

San, O. and Maulik, R. (2018a). Machine learning closures for model order reduction of thermal fluids. *Applied Mathematical Modelling*, 60:681–710.

San, O. and Maulik, R. (2018b). Neural network closures for nonlinear model order reduction. *Advances in Computational Mathematics*, 44(6):1717–1750.

Sarma, K. C. and Adeli, H. (2001). Bilevel parallel genetic algorithms for optimization of large steel structures. *Computer-Aided Civil and Infrastructure Engineering*, 16(5):295–304.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. (2013). Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, pages 2263–2291.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Shao, H., Jiang, H., Zhang, H., and Liang, T. (2017). Electric locomotive bearing fault diagnosis using a novel convolutional deep belief network. *IEEE Transactions on Industrial Electronics*, 65(3):2727–2736.

Sharma, R. K., Sugumaran, V., Kumar, H., and Amarnath, M. (2015). A comparative study of naïve bayes classifier and bayes net classifier for fault diagnosis of roller bearing using sound signal. *International Journal of Decision Support Systems*, 1(1):115–129.

Shi, M., Lv, L., Sun, W., and Song, X. (2020). A multi-fidelity surrogate model based on support vector regression. *Structural and Multidisciplinary Optimization*, pages 1–13.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Shrestha, D., Kayastha, N., and Solomatine, D. (2009). A novel approach to parameter uncertainty analysis of hydrological models using neural networks. *Hydrology and Earth System Sciences*, 13(7):1235–1248.

Shwartz-Ziv, R. and Tishby, N. (2017). Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

Si, S., Tao, D., and Geng, B. (2009). Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(7):929–942.

Singh, J., Azamfar, M., Ainapure, A., and Lee, J. (2020). Deep learning-based cross-domain adaptation for gearbox fault diagnosis under variable speed conditions. *Measurement Science and Technology*, 31(5):055601.

Singh, M. P. and Chowdhury, B. (2011). *Control of autonomous underwater vehicles*. PhD thesis.

Škvára, V., Pevnỳ, T., and Šmídl, V. (2018). Are generative deep models for novelty detection truly better? *arXiv preprint arXiv:1807.05027*.

Soares, R. (2011). Inductive transfer. *Encyclopedia of Machine Learning. Springer, Boston, MA*.

Sobol', I. M., Asotsky, D., Kreinin, A., and Kucherenko, S. (2011). Construction and comparison of high-dimensional sobol'generators. *Wilmott*, 2011(56):64–79.

Sohn, H., Dzwonczyk, M., Straser, E. G., Kiremidjian, A. S., Law, K. H., and Meng, T. (1999). An experimental study of temperature effect on modal parameters of the alamosa canyon bridge. *Earthquake engineering & structural dynamics*, 28(8):879–897.

Sohn, H., Farrar, C. R., Hemez, F. M., and Czarnecki, J. J. (2002). A review of structural health review of structural health monitoring literature 1996-2001. Technical report, Los Alamos National Laboratory, Los Alamos, New Mexico.

Sohn, H., Worden, K., and Farrar, C. R. (2001). Novelty detection using auto-associative neural network. In *2001 ASME International Mechanical Engineering Congress and Exposition*, pages 573–580.

Sreekanth, J. and Datta, B. (2010). Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models. *Journal of hydrology*, 393(3-4):245–256.

Stephens, D., Gorissen, D., Crombecq, K., and Dhaene, T. (2011). Surrogate based sensitivity analysis of process equipment. *Applied Mathematical Modelling*, 35(4):1676–1687.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J., and Nenadic, G. (2019). Machine learning methods for wind turbine condition monitoring: A review. *Renewable energy*, 133:620–635.

Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5).

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. (2008). Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440.

Sun, H. and Betti, R. (2015). A hybrid optimization algorithm with bayesian inference for probabilistic model updating. *Computer-Aided Civil and Infrastructure Engineering*, 30(8):602–619.

Sun, W., Chen, J., and Li, J. (2007). Decision tree and pca-based fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing*, 21(3):1300–1317.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Taigman, Y., Polyak, A., and Wolf, L. (2016). Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

Tang, B. (1993). Orthogonal array-based latin hypercubes. *Journal of the American statistical association*, 88(424):1392–1397.

Tang, S., Shen, C., Wang, D., Li, S., Huang, W., and Zhu, Z. (2018). Adaptive deep feature learning network with nesterov momentum and its application to rotating machinery fault diagnosis. *Neurocomputing*, 305:1–14.

Teughels, A. and De Roeck, G. (2005). Damage detection and parameter identification by finite element model updating. *Revue européenne de génie civil*, 9(1-2):109–158.

Thompson, M. L. and Kramer, M. A. (1994). Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal*, 40(8):1328–1340.

Tibaduiza, D. A., Mujica, L. E., and Rodellar, J. (2012). Damage classification in structural health monitoring using principal component analysis and self-organizing maps. *Structural Control and Health Monitoring*, 20(10):1303–1316.

Turner, C. J., Crawford, R. H., and Campbell, M. I. (2007). Multidimensional sequential sampling for nurbs-based metamodel development. *Engineering with Computers*, 23(3):155–174.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2015). Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pages 4068–4076.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

van der Herten, J., Couckuyt, I., Deschrijver, D., and Dhaene, T. (2015). A fuzzy hybrid sequential design strategy for global surrogate modeling of high-dimensional computer experiments. *SIAM Journal on Scientific Computing*, 37(2):A1020–A1039.

Van Der Herten, J., Deschrijver, D., and Dhaene, T. (2014). Fuzzy local linear approximation-based sequential design. In *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, pages 17–21. IEEE.

Van Der Herten, J., Van Steenkiste, T., Couckuyt, I., and Dhaene, T. (2017). Surrogate modelling with sequential design for expensive simulation applications. *Computer Simulation*, page 173.

Wang, J. and Jiang, J. (2019). Conditional coupled generative adversarial networks for zero-shot domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Wang, J., Wiens, J., and Lundberg, S. (2021). Shapley flow: A graph-based approach to interpreting model predictions. In *International Conference on Artificial Intelligence and Statistics*, pages 721–729. PMLR.

Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153.

Wang, S.-Q. and Li, H.-J. (2012). Assessment of structural damage using natural frequency changes. *Acta Mechanica Sinica*, 28(1):118–127.

Wang, T., Bhuiyan, M. Z. A., Wang, G., Rahman, M. A., Wu, J., and Cao, J. (2018). Big data reduction for a smart city's critical infrastructural health monitoring. *IEEE Communications Magazine*, 56(3):128–133.

Wang, Z., Lin, R., and Lim, M. (1997). Structural damage detection using measured frf data. *Computer methods in applied mechanics and engineering*, 147(1-2):187–197.

Widodo, A. and Yang, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, 21(6):2560–2574.

Wilding, M. (2019).

Willard, J., Jia, X., Xu, S., Steinbach, M., and Kumar, V. (2020). Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*.

Wilson, G. and Cook, D. J. (2020). A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46.

Wirsching, P. H., Paez, T. L., and Ortiz, K. (2006). *Random vibrations: theory and practice*. Courier Corporation.

Woon, C. E. and Mitchell, L. D. (1996). Variations in structural dynamic characteristics caused by changes in ambient temperature: I. experimental. In *Proceedings - SPIE The International Society for Optical Engineering*, pages 963–671. SPIE International Society for Optical Engineering.

Worden, K. (1997a). Damage detection using a novelty measure. In *Proceedings - SPIE The International Society for Optical Engineering*, pages 631–637. SPIE International Society for Optical Engineering.

Worden, K. (1997b). Structural fault detection using a novelty measure. *Journal of Sound and Vibration*, 201(1):85–101.

Worden, K. and Manson, G. (2006). The application of machine learning to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):515–537.

Worden, K., Manson, G., and Allman, D. (2003). Experimental validation of a structural health monitoring methodology: part i. novelty detection on a laboratory structure. *Journal of Sound and Vibration*, 259(2):323–343.

World Bank (2019). 2017 infrastructure report card. https://lpi.worldbank.org/international/global. Accessed: 2020-05-15.

Xi, W. (2014). *Performance Based Implementation of Seismic Protective Devices for Structures*. PhD thesis, UCLA.

Xia, M., Li, T., Liu, L., Xu, L., and de Silva, C. W. (2017). Intelligent fault diagnosis approach with unsupervised feature learning by stacked denoising autoencoder. *IET Science, Measurement & Technology*, 11(6):687–695.

Xia, Y., Chen, B., Weng, S., Ni, Y., and Xu, Y. (2012). Temperature effect on vibration properties of civil structures: a literature review and case studies. *Journal of Civil Structural Health Monitoring*, 2(1):29–46.

Xie, J., Du, G., Shen, C., Chen, N., Chen, L., and Zhu, Z. (2018). An end-to-end model based on improved adaptive deep belief network and its application to bearing fault diagnosis. *IEEE Access*, 6:63584–63596.

Xie, J., Zhang, L., Duan, L., and Wang, J. (2016). On cross-domain feature fusion in gearbox fault diagnosis under various operating conditions based on transfer component analysis. In *2016 IEEE international conference on prognostics and health management (ICPHM)*, pages 1–6. IEEE.

Xu, T. and Valocchi, A. J. (2015). Data-driven methods to improve baseflow prediction of a regional groundwater model. *Computers & Geosciences*, 85:124–136.

Xu, Y. L., Chen, B., Ng, C. L., Wong, K. Y., and Chan, W. Y. (2009). Monitoring temperature effect on a long suspension bridge. *Structural Control and Health Monitoring*, 17(6):632–653.

Yağdereli, E., Gemci, C., and Aktaş, A. Z. (2015). A study on cyber-security of autonomous and unmanned vehicles. *The Journal of Defense Modeling and Simulation*, 12(4):369–381.

Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., and Zuo, W. (2017). Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2272–2281.

Yao, H., Gao, Y., and Liu, Y. (2020). Fea-net: A physics-guided data-driven model for efficient mechanical response prediction. *Computer Methods in Applied Mechanics and Engineering*, 363:112892.

Yao, K., Herr, J. E., Toth, D. W., Mckintyre, R., and Parkhill, J. (2018). The tensormol-0.1 model chemistry: a neural network augmented with long-range physics. *Chemical science*, 9(8):2261–2269.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. (2019). Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32.

Yu, J., Bai, M., Wang, G., and Shi, X. (2018). Fault diagnosis of planetary gearbox with incomplete information using assignment reduction and flexible naive bayesian classifier. *Journal of Mechanical Science and Technology*, 32(1):37–47.

Yu, L., Zhu, J., and Cheri, L. (2010). Parametric study on pca-based algorithm for structural health monitoring. In *2010 Prognostics and System Health Management Conference*, pages 1–6.

Yu, Y., Wang, C., Gu, X., and Li, J. (2019). A novel deep learning-based method for damage identification of smart building structures. *Structural Health Monitoring*, 18(1):143–163.

Yuen, M. M. F. (1985). A numerical study of the eigenparameters of a damaged cantilever. *Journal of sound and vibration*, 103(3):301–310.

Zachariadis, I. A. (2018). Investment in infrastructure in the eu: Gaps, challenges, and opportunities. https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS_BRI(2018)628245. Accessed: 2020-05-15.

Zang, C. and Imregun, M. (2001). Structural damage detection using artificial neural networks and measured frf data reduced via principal component projection. *Journal of Sound and Vibration*, 242(5):813–827.

Zhang, J., Chowdhury, S., and Messac, A. (2012). An adaptive hybrid surrogate model. *Structural and Multidisciplinary Optimization*, 46(2):223–238.

Zhang, L. and Gao, X. (2019). Transfer adaptation learning: A decade survey. *arXiv preprint arXiv:1903.04687*.

Zhang, L., Wang, G., and Giannakis, G. B. (2018a). Real-time power system state estimation via deep unrolled neural networks. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 907–911. IEEE.

Zhang, R., Liu, Y., and Sun, H. (2020). Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling. *Engineering Structures*, 215:110704.

Zhang, W., Ouyang, W., Li, W., and Xu, D. (2018b). Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3801–3809.

Zhang, X., Xie, F., Ji, T., Zhu, Z., and Zheng, Y. (2021). Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 373:113485.

Zhang, Z. and Sun, C. (2021). Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating. *Structural Health Monitoring*, 20(4):1675–1688.

Zhao, H., Des Combes, R. T., Zhang, K., and Gordon, G. (2019). On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR.

Zhao, M., Kang, M., Tang, B., and Pecht, M. (2017). Deep residual networks with dynamically weighted wavelet coefficients for fault diagnosis of planetary gearboxes. *IEEE Transactions on Industrial Electronics*, 65(5):4290–4300.

Zhao, M., Kang, M., Tang, B., and Pecht, M. (2018). Multiple wavelet coefficients fusion in deep residual networks for fault diagnosis. *IEEE Transactions on Industrial Electronics*, 66(6):4696–4706.

Zhou, H. F., Ni, Y. Q., and Ko, J. M. (2011). Eliminating temperature effect in vibration-based structural damage detection. *Journal of Engineering Mechanics*, 137(12):785–796.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Zhuang, F., Cheng, X., Luo, P., Pan, S. J., and He, Q. (2015). Supervised representation learning: Transfer learning with deep autoencoders. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.