A Data-Driven Perspective on Residential Electricity Modeling and Structural Health Monitoring

Lechen Li

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2023

ABSTRACT

A Data-driven Perspective on Residential Electricity Modeling and Structural Health Monitoring

Lechen Li

In recent years, due to the increasing efficiency and availability of information technologies for collecting massive amounts of data (e.g., smart meters and sensors), a variety of advanced technologies and decision-making strategies in the civil engineering sector have shifted in leaps and bounds to a data-driven manner. While there is still no consensus in industry and academia on the latest advances, challenges, and trends in some innovative data-driven methods related to, e.g., deep learning and neural networks, it is undeniable that these techniques have been proved to be considerably effective in helping our academics and engineers solve many real-life tasks related to the smart city framework. This dissertation systematically presents the investigation and development of the cutting-edge data-driven methods related to two specific areas of civil engineering, namely, Residential Electricity Modeling (REM) and Structural Health Monitoring (SHM). For both components, the presentation of this dissertation starts with a brief review of classical data-driven methods used in particular problems, gradually progresses to an exploration of the related state-of-the-art technologies, and eventually lands on our proposed novel data-driven strategies and algorithms. In addition to the classical and state-of-the-art modeling techniques focused on these two areas, this dissertation also put great emphasis on the proposed effective feature extraction and selection approaches. These approaches are aimed to optimize model performance and to save computational resources, for achieving the ideal characterization of the

information embedded in the collected raw data that is most relevant to the problem objectives, especially for the case of modeling deep neural networks. For the problems on REM, the proposed methods are validated with real recorded data from multi-family residential buildings, while for SHM, the algorithms are validated with data from numerically simulated systems as well as real bridge structures.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Now it is time to thank those who have helped me along the path of my doctoral studies. I cannot imagine how I would have been able to complete this steep and lonely journey without those who have continued to support and inspire me unconditionally over the past four to five years.

I would like to thank my advisor, Prof. Raimondo Betti. Since I met him in 2016, he has been a great help and support to my research and spiritual life. In my mind, he has long since gone far beyond the role of an advisor and is more like a dear family elder, as well as my confidant. I would like to thank my other advisor, Prof. Patricia Culligan, who generously provided me with funding during the initial years of my Ph.D. and introduced me to the world of urban energy modeling. I am really grateful for the great patience she showed with my early mistakes and discomfort in my research, and for the huge faith she placed in my imagination to try out various approaches. I would also like to acknowledge my dissertation defense committee, Professors Raimondo Betti, Patricia Culligan, George Deodatis, Hoe I. Ling, and Richard Longman, for their time and helpful comments on my work.

I would like to thank Professors Christoph Meinrenken and Vijay Modi for their continuous and inspiring feedbacks on my research in energy modeling and data-driven algorithms, and for their great guidance in the conceptualization and writing of related papers.

I would like to thank Dr. Marcello Morgantini and Dr. Eleonora Maria Tronci for their endless help and support in my research on structural health monitoring, both in terms of theoretical analysis and experimental manipulation methods. I would also like to express my gratitude to all the faculty and staff of the Department of Civil Engineering and Engineering Mechanics.

Who would have thought that during my years at Columbia, I would witness that this city suddenly went from a place of hustle and bustle to a desperately cold one when the pandemic hit. During that time, one by one, many of my friends helplessly left this place. I still remember standing in the middle of empty Broadway in 2020, as if I could hear the slow and feeble heartbeat of this city. When everything got back on track in the fall of 2021, it felt like there was hope for everything again. Luckily, I have always had some brothers, also Ph.D. candidates, Wenxi Li, Yu Yang, Tianxu Lan, and Yifei Zong, to accompany me on this journey. I don't know how I would have gotten through these four long, lonely years without them.

At last, I would like to thank my parents, who have supported all my decisions unconditionally from the beginning to the end and have given me so much strength and hope to go through the most difficult period of my life so far.

Chapter 1. Introduction

The explosive development of big data and artificial intelligence have fully integrated the associated techniques into almost every aspect of our lives, and countless academic researchers and industrial engineers have been constantly exploring their new possibilities in a wide range of industries. In civil engineering, many topical problems, such as urban energy distribution, infrastructure system monitoring, and related technical management decisions, involve many uncertainties and complex theories whose solutions require not only mathematical and physical knowledge, but are also highly dependent on the experiential expertise from practitioners [1, 2]. However, such experiential expertise can be illogically incomplete and imprecise, and it sometimes cannot be systematically summarized with effective implementation through traditional operational processes due to considerable labor and time costs [1]. Fortunately, benefiting from the explosive growth of available data and computing resources as well as the huge advances in Artificial Intelligence (AI), data-driven methods such as Machine Learning (ML) algorithms, have evolved significantly in a range of engineering and science fields with clear advantages in overcoming the limitations of many traditional approaches. A major benefit is that they can solve complex problems by emulating experienced practitioners, i.e., through an effective learning process to reach the level of experts and thus gain the required expertise [1, 3].

As the name suggests, a data-driven method proposes a modeling framework through a data analysis scheme rather than simply following a classical physical or mathematical modeling manner, and today many scholars have developed diverse strategies to incorporate information of mathematical and physical principles into the data-driven mindset [4]. In recent years, these frameworks can be integrated with many advanced numerical analytic techniques, benefiting from substantial developments in areas such as digital/statistical signal processing and numerical

methods, allowing for great flexibility and effectiveness in modeling strategies. Data-driven methods can often be implemented fast with strong effectiveness and do not require great user expertise, related to one of the main reasons why data-driven strategies have recently become quite attractive to many scholars and engineers [3]. In addition, recent incredible advances in various sensors and computer hardware technologies have not only greatly facilitated the implementation of data acquisition and data fusion, but also ensured continuous computational resources for complex data-driven computational architectures, such as deep neural networks [5]. As a result, supported by these promising advances in the big data community, constantly increasing civil engineering scholars and engineers have commenced research and engineering implementations through data-driven methods that provide new case studies, algorithms, and results, while many technical challenges still remain [6].

1.1 Dissertation overview

In this dissertation, cutting-edge data-driven methods have been systematically explored and developed for two specific areas in civil engineering, namely Residential Electricity Modeling (REM) and Structural Health Monitoring (SHM). These two areas encompass many topical problems that are of broad interest to both academia and industry in civil engineering today, and have attracted large numbers of scholars and engineers over the past decade to explore a variety of advanced data-driven strategies to address related challenges.

For the REM, the recent significant growing interest in this topic is mainly due to the emergence of advanced smart grid technologies and various related application scenarios [7, 8]. In recent years, the penetration of smart meters has grown significantly and they are now becoming more widespread globally. Companies such as Google, Siemens, Intel, General Electric and Amazon have been developing end-use applications for setting up household battery-energy

management and load control systems based on smart meter data [8]. As the result of this progress, electricity utilities, governments and scholars have realized the appreciable benefits of using home smart meter data in energy efficiency improvement and demand response operations, since the high resolution and quality smart meter data can be a sufficient resource for residential electricity analysis and predictive modeling, alongside the data of useful exogenous variables such as weather condition recordings [9]. Motivated by these facts, data-driven method development for the field of REM has become one of the most prevalent topics related to building energy management in the era of smart grid technologies [8, 10].

For the SHM, it is gaining impressive attention in recent years as ensuring life safety and reducing inspection costs have become top priorities for practicing engineers and researchers [11, 12]. Moreover, recent advancements in sensor and communication technologies (contact and contactless, wired and wireless, etc.) have created great opportunities for the acquisition of observational data at an incredible rate and amount, which have laid solid foundation for the large-scale development and application of data-acquisition techniques and data-driven methods in SHM, leading to promising benefits to minimize the direct and indirect money and labor costs associated with the streamline periodic inspections for aging infrastructure [6]. Traditionally, SHM solutions tend to land on building physical models (e.g., a finite element model) to represent the dynamic characteristics of a real structural system, but such models typically require that the actual data measured have minimal noise, and also require precise control and understanding of the details for the model parameters, which may result in undesirable stability of the model performance due to the interference from the varying environmental conditions [12]. In contrast, data-driven models, by virtues of the large amount of heterogeneous data from sensors, can efficiently and consistently provide bottom-up solutions including diagnostics and prognostics (e.g., damage detection and

remaining life estimation) through their natural strengths in big data mining and interpretation [13]. Consequently, data-driven modeling nowadays has become one of the most attractive strategies for SHM problems [6].

1.2 Similarities and differences of the two areas in a data-driven perspective

In terms of data forms and study paradigms, a wide range of classical and state-of-the-art research on data-driven methods for the two areas is based on the acquisition and analysis of dynamic data with time as the independent variable (i.e., the time-series data), with specific operations including data wangling and cleaning, feature extraction and selection, algorithm development and validation, etc. [7, 13]. Either smart meters placed in buildings to record residential electricity usage or various sensors placed on monitored structural systems to measure structural response data can provide scholars and engineers with adequate and comprehensive time-series data to solve relevant problems or validate their developed methods. In this dissertation, in addition to presenting the newly developed modeling strategies for the considered problems in the two areas, extensive investigations are conducted on the key intrinsic factors affecting the modeling performance with respect to general time-series variables. Furthermore, the different practical engineering challenges of the two areas presented in this dissertation show what flexibility and additional requirements should be met when implementing data-driven modeling in the face of evolving real-world scenarios.

For residential electricity usage records, these time series data can be regarded as relatively normal scalar data generated in daily life. Therefore, the analysis and modeling of this data can be addressed by standard data-driven strategies, either through classical statistical models or through cutting-edge machine learning techniques. The essential objective in this case is to maximize the performance (accuracy) of fitting the data, such as the R-squared value in regression problems,

while ensuring the robustness and generalization of the model. Therefore, for the problems in REM, scholars have tended to dedicate more efforts to drilling down the validity, precision, and innovation points of the developed models in terms of data-fitting performance or to conducting in-depth statistical analysis/inference on the information about electricity load profiles and resident daily behaviors [7].

However, for the structural response data in SHM problems, the data-driven models developed should achieve a certain accuracy of data fitting while meeting the requirements of consistency with specific physical/mathematical principals followed by the response data. Taking the acceleration response of structural vibrations as an example, the modeling process must take into account how effectively the built model perceives the physical properties underlying the response, such as the natural frequencies and damping ratios of a monitored structural system, so that the model can be trained to provide a well-defined physical description of the monitored system. Therefore, this requires that the model should be highly interpretable and not like a black box where there are some data predictions/fittings that cannot be clearly explained by mathematical or physical theory (even though the resulting accuracy values may be high). This is why, in recent years, a growing number of scholars have worked on developing physics-informed machine learning techniques or novel feature engineering strategies to effectively simplify/modify complex structures (e.g., the deep neural networks), which allow the developed models to be not only functionally more stable and efficient, but also more interpretable at the theoretical level [6, 13].

1.3 Logics and organization of this dissertation

As discussed above, research on data-driven methods for the problems in REF and SHM has yielded a number of progressive results over the past few decades. Therefore, the presentation framework of each chapter in this dissertation generally starts with investigating or experimenting

with the existing relevant data-driven methods for target problems, then explores the advantages and limitations of these methods through a systematical comparison, and finally presents newly developed methods or modified existing methods that are proposed to improve problem-solving performance, with various case studies conducted for the corresponding validation. With the purpose of optimizing the modeling process, this dissertation also highlights the effective strategies based on feature engineering to extract the most relevant information embedded in the raw measurement data to better address some common modeling challenges in the two areas. These strategies can usually support a more concise and rational architecture of the models in the case of developing complex deep neural networks, thus allowing them to gain considerable generalization and interpretation capabilities.

The content of REF is presented in Chapters 2 – 3: In Chapter 2, a typical and widely followed REF problem, termed as short-term load forecasting [14], is systematically researched through a case study on the electricity usage records from a residential electricity database in New York City (NYC). An in-depth investigation of classical and up-to-date data-driven methods to the short-term load forecasting problem is first presented. To maximize the exploitation of autocorrelation information in the time-series electricity data to perform the forecasting better, a novel modeling strategy is proposed, which consists of an improved recurrent neural network framework, a newly developed dynamic feature selection algorithm, and a "default" state configured within the model to address overfitting issues.

In Chapter 3, a further analysis of the recorded electricity usage (from the same database considered in Chapter 2) under a special situation is performed, i.e., how the residential electricity consumption and load demand would change under the impact of the Covid-19 related lockdown, and what the potential threat to the grid is from the extreme peak load demand during this period.

A series of analytical methods based on mathematical modeling and statistical simulation are proposed, while significant factors affecting the changes in the load profiles are deeply analyzed and then considered as the key predictors for the modeling. The simulated scenarios with corresponding forecasting results drawn from the developed models can serve as timely alerts for electricity utilities and provide informative insights into future grid stability.

The content of SHM is presented in Chapters 4 – 6: In Chapter 4, it focuses on the development of data-driven methods to address the problem of structural damage detection and quantification. An overview of the structural damage assessment in a vibration-based SHM framework is presented first, including a brief literature review of some classical modeling strategies, different types of Damage Sensitive Features (DSFs), and relevant state-of-the-art data-driven techniques emerged recently. To improve the accuracy and robustness of a Traditional Auto-Encoder (TAE)-based modeling method for structural damage detection, a New Generalized Auto-Encoder (NGAE) architecture, integrated with the power cepstral coefficients of acceleration response and a statistical-pattern-recognition strategy, is then proposed. The proposed NGAE architecture is able to be well-generalized in the necessary structural physical properties of a target system thanks to a newly defined encoder-decoder mapping, finally resulting in excellent damage detection and quantification performance.

In Chapter 5, motivated by the objective to recognize various damage scenarios rather than just detecting the presence of damage as in Chapter 4, the problem of structural damage classification is studied. To address the problem properly, a novel data augmentation strategy based on a Conditional Variational Autoencoder (CVAE) architecture is proposed to create a "balanced" training dataset of the cepstral coefficients for various structural undamaged and damaged conditions. This augmented training dataset of the cepstral coefficients can be employed

to better train a Probabilistic Linear Discriminant Analysis (PLDA) model to finally achieve greater accuracy and robustness in structural damage classification, compared to using the original "unbalanced" training dataset.

In Chapter 6, following the analysis and findings in Chapters 4 and 5, the problems of structural damage localization for linear and nonlinear structural systems are systematically studied. By utilizing the properties of the Linear Discriminant Analysis (LDA), a novel data-driven method is proposed to address the structural damage localization problem in an unsupervised-learning manner. The key intuition of this method is to highly extract and exploit the structural local characteristics embedded in the cepstral coefficients based on the strong capability of separating categorical data offered by the mechanism of the LDA.

The last part of the dissertation (Chapter 7) sums up the findings and contributions of the research, while identifying possible streams of future research.

Chapter 2. Short-term Load Forecasting in Multi-family Residential Buildings

The main part of this chapter is presented in the paper co-authored with Prof. Christoph Meinrenken, Prof. Vijay Modi and Prof. Patricia Culligan, and published in the Journal of Applied Energy [15].

2.1 Introduction

In recent years, residential electricity load profiles have become increasingly varied among neighborhoods and homes due to modified work and leisure patterns, increased use of electronics, and more frequent presence of distributed generation (e.g., roof top photovoltaic) and storage (e.g., electric vehicles) [16]. This increases the benefit of and need for electrical networks such as Transactive Energy Networks (TENs) [16], which could transform homes from being a passive load into a smart storage and demand responsive entity for electric grids, thus enabling a dynamic balance of demand and deeper integration of emerging clean electricity generation. For example, Zheng et al. [17] introduced a model for levelized storage cost, based on storage lifetime and electricity tariffs, and developed a storage dispatch algorithm to optimize the storage size and the grid demand limits. Similarly, as reviewed by Song et al. [18], a host of novel market mechanisms and respective technology solutions are under consideration to improve the resiliency and reduce the carbon intensity of electricity grids. Most of these innovations will either require, or at least benefit from, the ability to forecast short-term electricity consumption patterns at the level of individual actors in a TEN (with “short-term” typically referring to 30 min to one-week time periods, but not longer) [14]. In the case of multifamily residential buildings, which are common

in many urban areas around the world, the level of an individual actor could include an individual apartment, a floor, or an entire apartment building, for example.

Implementation of such intelligent and adaptive elements requires advanced techniques for accurate and precise load demand and power generation forecasting. For short-term load forecasting, many approaches have been studied but few have focused on the electricity load of individual households, for two reasons: First, electricity load profiles of individual households can reveal private information that often cannot be published, contributing to a lack of data availability for the residential sector, especially in multi-family residential buildings. Second, forecasting the electricity load of individual households is conventionally considered challenging due to the volatile nature of household load data [19].

A large portion of the existing work on electricity use forecasting has focused on commercial buildings due to the availability of datasets and the often more easily identifiable diurnal use patterns (reviewed in, e.g., Meinrenken and Mehmani [20]). For residential buildings, researchers have developed various statistical models and machine learning algorithms for load prediction. Many of them used datasets containing only one level of spatial aggregation (e.g., the aggregate load profile of an entire building). A few studies have carried out comparative experiments based on various scenarios to investigate the influence of different forecast granularities (e.g., load at level of individual apartments or their aggregates at floor or building level, and load at level of hourly, daily, or weekly time intervals), and some other studies have aimed at improving forecasting accuracy by overcoming some common challenges of machine learning algorithms such as overfitting (reviewed by Amasyali et al. [21]).

2.1.1 Load forecasting models for residential electricity use

As noted, although many studies have focused on the forecasting of electricity load in residential buildings, only a few of them have been conducted on individual households [7]. One such study, by Ghofrani et al. [22], forecasted the electricity load of one specific household. A Kalman Filter estimator was applied, and the load was forecasted hourly and sub-hourly as a sum of two separate components: the weather-dependent component and the lifestyle component. The authors used mean absolute percentage error as the accuracy metric and obtained forecasting accuracies between 18% and 30%. Munkhammar et al. [23] employed what is referred to as a "Markov-chain mixture distribution model" to forecast one step ahead (half-hour resolution) residential electricity consumption data from Australia.

Previous studies also addressed the problem of identifying an optimal model for residential load forecasting tasks by comparing the accuracies of various machine learning algorithms. For example, Edward et al. [9] implemented seven different models, including multiple linear regression, support vector machine, and deep neural networks, to forecast one-hour ahead electricity loads of a residential building. While the tested models showed reliable forecasts when considering the average coefficient of variation (CV), compared to similar work, their datasets were limited to only three individual households. Therefore, the need remains to validate such models on larger datasets of measured electricity consumption.

In recent years, deep learning models have been shown to offer many advantages, and they often perform better than traditional machine learning. Both Zheng et al. [24] and Marino et al. [25] have succeeded in applying a Long Short-Term Memory (LSTM) Neural Network to short-term load forecasting in residential buildings. They concluded that the LSTM neural network has an advantage in handling data-driven electricity consumption forecasting tasks. Andriopoulos et

al. [26] applied a Convolutional Neural Network (CNN) to a short-term load forecasting task for three individual households. They employed a statistical analysis to convert their original dataset to a format that facilitated leverage of the advantages of the CNN algorithm. They concluded that the proposed CNN can outperform conventional LSTM in cases where the number of data observations are limited (such as the loads in a small energy community) and the load patterns change dynamically.

2.1.2 Forecasting at different spatial and temporal scales

In addition to modeling techniques, the spatial and/or temporal scale of forecasting (or sometimes granularity [27]) is another factor affecting the forecasting accuracy. Electricity load data in multi-family residential buildings, for example, can be obtained at varying temporal granularity such as 15 mins, 1 h, or 1 day, and at different spatial granularities such as household, floor, or building level. Determining the optimal forecast granularity is an important aspect of improving accuracy of the forecast.

Determining the optimal spatial and temporal granularities at the same time, Jain et al. [27] applied a Support Vector Regression (SVR) model to make one-step load predictions for a residential building at 10- min, hourly, and daily temporal granularities, as well as household, floor, and building spatial granularities. They found the optimal forecasting granularity to be for one-hour ahead and at floor level. Zheng et al. [28] developed a Kalman filter-based bottom-up method to increase the accuracy of household-load forecasting. They verified the advantages of this approach via granularity analysis at the level of appliances, rooms, and household, and found that the Kalman filter bottom-up method at the appliance level can improve household load forecasting accuracy. Xu et al. [29] applied a probability-based electricity forecasting model for

buildings that decomposed the load into a baseline load and an abnormal peak load. They concluded that such a decomposition technique can provide more granular data for forecasting models and hence increase forecasting accuracy.

2.1.3 Feature selection and sparse models

Sparse models and feature selection techniques have been shown to improve electricity load prediction by capturing certain key features [30]. Therefore, these approaches could be utilized to obtain a generalized model by lowering the risk of overfitting, as they can focus on a small amount of core information highly correlated with electricity use.

Regarding the sparse coding techniques used in existing literature, Jain et al. [31] applied a lasso regression model, which is a shrinkage and selection approach to linear regression that approximates sparse coefficients, to forecast energy use in an NYC multifamily residential building. They concluded that the lasso regression model provides competitive performance compared with a support vector machine. Candanedo et al. [32] presented a data-filtering method by removing non-predictive parameters and unrelated features, to improve the performance of 4 statistical models for the energy use of appliances in a low-energy house. With the method, they concluded that the gradient boosting machines (GBM) outperformed the other 3 used models, which achieved the accuracy of 57% in R^2.

More recently, with respect to forecasting electricity load and other time-series data, some studies have implemented feature selection by integrating advanced deep learning techniques. Amarasinghe et al. [33] developed a 1-D convolutional neural network (1D-CNN) performing energy load forecasting at individual building level. Their experimental results showed that the CNN outperformed SVR. However, using such an approach here is unlikely to succeed when

temporal load profiles have large seasonal volatility and unexpected load changes due to human behaviors. Regarding the recently used recurrent neural networks, Wang et al. [34] developed a novel short-term load forecasting method based on the attention mechanism (AM), rolling update (RU) and bi-directional long short-term memory (Bi-LSTM) neural network. When comparing the Bi-LSTM model with AM and RU to a traditional Bi-LSTM model, both the mean absolute percentage error (MAPE) and the root mean square error (RMSE) were shown to decrease in the load forecasting associated with their two data sets. Wan et al. [35] employed a temporal convolutional network, integrated with encoder-decoder layers by using a sequence-to-sequence (Seq2Seq) framework, to yield better hidden representation of features for time-series data forecasting. They concluded that their developed architectures outperform many multivariate regressions techniques.

2.1.4 Focus of this chapter and differentiation from previous work

The above-mentioned studies, using sparse coding techniques and advanced neural network techniques, usually automatically obtain the most influential hidden feature representation by using fixed types of features. However, this approach usually does not consider whether these features are always dominant under different situations such as seasonal changes and some idiosyncratic human behaviors. Similarly, although most of the above discussed studies applied past electricity load values as important features for prediction, they did not consider dynamic methods of continuously updating the selection to the most correlated feature types in order to enhance the feature representation before feeding them to the forecasting models. Such a dynamic feature selection process is needed when electricity consumption of an apartment (as in our dataset described in Section 2.2.1) could be primarily due to loads from refrigerators and standby-mode

electronics that are present even when a resident is not in the apartment for up to several days. This can lead to overfitting if this problem is addressed by using multiple previous time-step load values (as done by the aforementioned studies) and by relying only on the feature-selection process of the encoder-decoder layers of Seq2Seq and TCNN or the sum of weighted states of AM-based LSTM structure. Aiming to address these issues, we extend a previously introduced Convolutional LSTM framework (ConvLSTM) [36], whose built-in kernels allow the extraction of key information, by adding a dynamic feature-selection algorithm and a model-simplification approach, which enables timely reactions to the rapidly changing states of various load profiles in case of overfitting. The resulting ConvLSTM–based neural network with selected autoregressive features (henceforth CLSAF model) is tested as a short-term load prediction in a multifamily residential setting over three different season types (winter, summer, and the shoulder seasons of spring or fall) and across three spatial granularities (apartment, floor, and building level).

To test the feasibility and forecasting performance of our approach, we use a residential apartment building in New York City, NY, USA as a case study. We use the actual, hourly apartment-level electricity load of 59 individual apartments across 11 floors and from three different seasons (2019 data) to train the forecasting models and evaluate their accuracy. This data-rich case study allows us to systematically evaluate the effects of season, spatial granularity, and model choice on the forecasting accuracy. Finally, we determine two key characteristics of the residential load data and how these affect the forecasting accuracy for different apartments or floors. Based on this analysis, we discuss basic elements of a possible data screening technique, which could aid in providing confidence levels of load predictions to facilitate more complex transaction schemes within TENs.

2.2 Data and methods

2.2.1 Overview of electricity data

The considered electricity use dataset in this work is the historical electricity consumption records of a pre-1940 multi-story residential building in Manhattan, NYC, from an electricity database named MFRED (a detailed description of the database can be referred to [37].). The building is a pre-war construction with a steam-based, central heating system and electric window air conditioners for cooling. Therefore, air conditioning loads are reflected in the apartments' electricity use, whereas heating loads are not (except for the occasional supplementary heating via personal electric space heaters or heating blankets, for example). Electricity use for every apartment was separately metered by a Siemens® SEM3 micro-meter system with 50-amp split core current transformers and ±1% accuracy. As the model training data at apartment-level, we used the incremental electricity consumption (kWh) from one hour to the next. For the floor and building level, we first aggregated the observed electricity load of the respective apartments at either the floor or building level, and then used the aggregated data as training data to forecast the aggregate level.

The dataset contains 59 individual apartments, eleven floors, and one building, for three time periods in 2019, to reflect various weather conditions during the year: a period in winter when the use of indoor lights and possible auxiliary use of electric space heaters is highest (Jan. 7th to Feb. 3rd); a period during a shoulder season when little or no auxiliary heating but also little or no air conditioning will be used (Apr. 1st to Apr. 28th); and a period in the summer when the use of air conditioning is high (July 15th to Aug. 11th). In order to ensure comparability of the 3 different time periods, each period was chosen to start on a Monday and to last exactly 28 days, such that the different periods would each comprise of the same number of weekdays and weekend days.

For convenience, we henceforth refer to these three periods simply as ‘January’, ‘April’ and ‘July’, respectively.

Figure 2.1 (a) displays the diurnal load profiles averaged over all 59 apartments. **Figure 2.1 (b)** shows three examples of the hourly consumption of individual apartments during a one-day period. Data averaged over all apartments show systematic load patterns (e.g., high in the evening hours, low during the night), whereas some individual apartments do not, with volatile loads, partially caused by residents leaving the apartment for several days at a time. As reviewed in the Introduction, such idiosyncratic patterns render load forecasting more challenging. In response to such challenges, the model approach developed here aims at extracting the most correlated information from daily load profiles as prediction features, in order to mitigate the interference of idiosyncratic human behavior with prediction accuracy.

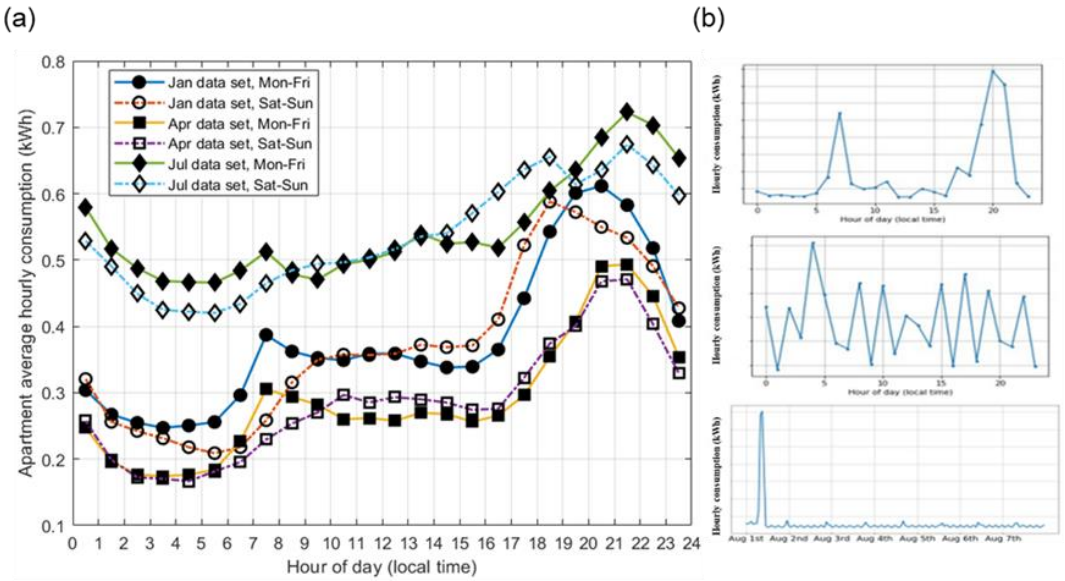

Figure 2.1: (a) Diurnal patterns of average hourly electricity use in 59 apartments. (b) Example daily electricity-load profiles of three sample apartments.

2.2.2. Metric to evaluate forecasting accuracy

In past studies, four types of metrics have been widely used to assess forecasting accuracy [38]: (i) scale-dependent measures, e.g., Root Mean Square Error or Mean Absolute Error; (ii) normalized metrics, e.g., Mean Absolute Percent Error (MAPE) or Coefficient of Variation (CV); (iii) relative metrics such as Mean Relative Error; and (iv) scale-free metrics such as Mean Absolute Scaled Error. Since the scale-dependent measures cannot be used for comparing the accuracy of forecasting at different magnitudes and MAPE is not applicable when handling the case of zero load values, the CV is considered in this work, as applied, e.g., by Jain et al. [27]. We refer to CV as CV-residual, in order to distinguish it from another, similarly defined metric in the following sections. CV-residual is defined as follows:

$$CV_{residual} = \frac{\sqrt{\frac{1}{N-1}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}}{\bar{y}} \tag{2.1}$$

where N is the number of individual hourly load observations for which the load is forecasted. In our study, N is equal to 504 $(24 \times 7 \times 3)$, representing the hourly load over the last three weeks of each 4-week time period (the first week is used for training and the last three weeks are used for testing, as shown in **Table 2.1**). y_t and $\hat{y}_t$ are the observed and predicted hourly load at time step t, respectively. y is the mean value of the N observations of the hourly electricity load.

2.2.3 Forecasting models and features used

In this work, we firstly try 4 benchmark models and a ConvLSTM model to complete the forecasting task. Then, by overcoming some disadvantages of the ConvLSTM model, and combining it with some advantages of one benchmark model, a more accurate and robust CLSAF model is developed, which can carry out short-term load forecasts for all scenarios (i.e., for the

three spatial granularities and three seasons). **Table 2.1** provides an overview of the 6 models and the corresponding feature types used in the present study.

For the source and accuracy of the electricity data, please refer to Section 2.2.1. All weather data (i.e., both for training and for testing) are historical 2019 data as shown in **Table 2.1**, which was obtained from the National Oceanic and Atmospheric Association (NOAA), NY Central Park Station. The temperature data is accurate to ±0.3 ˚C, and the humidity and wind speed data to around ±1%. In practical applications of the forecasting, the weather conditions used as the exogenous features for the forecasting of each time-step will be the one-hour weather forecasts (however, for the initial training ("warm-up") period, the models would still use actual, observed weather conditions, as shown in **Table 2.1**).

Regarding the use of exogenous features shown in **Table 2.1**, dry-bulb temperature (henceforth "temperature"), absolute humidity (henceforth "humidity"), wind speed, binary weekday/weekend, and the sinusoid of local time were chosen as our predictors. Wind speed rather than wind direction was chosen as one of the features as the prevailing wind direction at a weather station is not indicative of the actual wind direction at a specific apartment in a dense urban setting. On the other hand, wind speed and solar radiation are more likely to be closely associated with cooling and lighting needs at the apartment in question and have been used in our analyses. However, when solar radiation was added as an additional exogenous feature, it was detrimental to accuracy; this may be because in multi-family high rise buildings, only the predominantly south facing apartments or the apartments on the higher floors are subjected to direct solar radiation, even if the sun shines. Therefore, we decided to remove irradiance from the list of exogenous features.

Table 2.1: Overview of the inputs, outputs, and training (warm-up) periods for the employed 6 models (4 benchmark models and 2 newly employed models).

Model name	Autoregressive features (hourly electricity load)	Exogenous features (hourly granularity)	Training (warm-up) period	Output (one-step ahead hourly electricity load)
Persistence model	y[t-1]	None	None	y[t]
ARIMA model	Selected by the default setting of "forecast" package in R	None	First 7 days of each 28-day period	y[t]
ETS model	Selected by the default setting of "forecast" package in R	None	First 7 days of each 28-day period	y[t]
SVR model	y[t-1]	Temperature[t], absolute humidity[t], wind speed[t], weekday/weekend[t] and sin(local time[t])	First 7 days of each 28-day period	y[t]
ConvLSTM model	y[t-1]	Temperature[t], absolute humidity[t], wind speed[t], weekday/weekend[t] and sin(local time[t])	First 7 days of each 28-day period	y[t]
CLSAF model	Selected $y[t-p_t]$ or y[t-1] (p_t denotes a selected lag from time index)	Temperature[t], humidity[t], wind speed[t], weekday/weekend[t] and sin(local time[t])	First 7 days of each 28-day period	y[t]

2.2.3.1 Benchmark models

As mentioned early, 4 classical benchmark models, namely a Persistence model [28], an Auto Regressive Integrated Moving Average (ARIMA) model [39], an Exponential Smoothing (ETS) model [39], and a Support Vector Regression model [27], are presented in this chapter. Their fundamental mechanism and applications to short-term load forecasting problems are introduced in the following.

Persistence models can be applied as a benchmark for time-series prediction applications [44]. As we aim at conducting a single-step hourly forecast, the persistence model we employ uses the hourly load observed during the most recent time step:

$$Single\ step\ forecasting: \quad \widehat{y_t} = y_{t-1} \tag{2.2}$$

where $\widehat{y_t}$ is the predicted hourly load at time step t and y_{t-1} is the observed hourly load at time step $t-1$.

ARIMA models and ETS models are two strong, and well-established model types for time-series forecasting [39]. Therefore, they were selected among our series of benchmark models and used on all the aforementioned datasets, with default parameters automatically selected by using the "forecast" package in R [40]. In addition, as discussed in Introduction, SVR models have proven to be a well-performing technique in residential load forecasting, so it was also selected as a benchmark model, using the same features as the ones for the employed ConvLSTM model to set up a SVR model, as shown in **Table 2.1**.

2.2.3.2 Convolutional long short-term memory neural network (ConvLSTM) model

Long Short-Term Memory (LSTM) Neural Networks have been proven to be an efficient and powerful approach to short-term residential load forecasting tasks across multiple spatial granularities, as shown, e.g., by Zheng et al. [24]. However, an LSTM model might not completely meet the requirements of our dataset, for two reasons: First, as shown in Section 2.2.1, electricity loads in some apartments are volatile without clear diurnal patterns. Second, the primary factors driving electricity load may vary between seasons. In particular, the ambient temperature will likely affect the electricity consumption of air conditioners during the summer time but will be less relevant in wintertime when a building is centrally heated. Consequently, it might be best to

only use the exogenous features most correlated with electricity load. For our forecasting task, we tried a ConvLSTM layer to capture core information of the exogenous features that are highly correlated with electricity load, by taking advantage of the built-in kernels.

The ConvLSTM neural network, introduced by Xingjian et al. [38], is a variant of the LSTM neural network, which integrates a convolution operation into the LSTM cell. The convolution operation takes the place of a matrix multiplication at each of the LSTM cell's gate, and thereby captures inherent spatial features by several convolution operators in multi-dimensional data. Xingjian et al. [36] applied their proposed ConvLSTM network to better capture the spatiotemporal correlations of their spatial data. They concluded that the ConvLSTM network outperforms an LSTM with fully connected layers for precipitation nowcasting.

A ConvLSTM cell consists of a series of operations that can store temporal information with a selection process by the built-in kernels, and timely erases the cell's memory, like an LSTM cell, to prevent gradient vanishing [49]. Fig. 4 displays the basic mechanism of a ConvLSTM cell whose operations can be formulated as six core equations:

$$
\begin{aligned}
\boldsymbol{f}_t &= \sigma\left(\boldsymbol{W}_{fx} * \boldsymbol{I}_t + \boldsymbol{W}_{fh} * \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) \\
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_{ix} * \boldsymbol{I}_t + \boldsymbol{W}_{ih} * \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \\
\widehat{\boldsymbol{C}}_t &= tanh(\boldsymbol{W}_{Cx} * \boldsymbol{I}_t + \boldsymbol{W}_{Ch} * \boldsymbol{h}_{t-1} + \boldsymbol{b}_C) \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_{ox} * \boldsymbol{I}_t + \boldsymbol{W}_{oh} * \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \\
\boldsymbol{C}_t &= \widehat{\boldsymbol{C}}_t \cdot \boldsymbol{i}_t + \boldsymbol{C}_{t-1} \cdot \boldsymbol{f}_t \\
\boldsymbol{h}_t &= tanh(\boldsymbol{C}_t) \cdot \boldsymbol{o}_t
\end{aligned}
\tag{2.3}
$$

where '•' denotes the Hadamard product and '$*$' the convolution operation. $\boldsymbol{I}_t$ is the input of the ConvLSTM cell at time step t. $\boldsymbol{h}_{t-1}$ and $\boldsymbol{C}_{t-1}$ are the output and state of the ConvLSTM cell at time step $t-1$, respectively. Similarly, $\boldsymbol{h}_t$ and $\boldsymbol{C}_t$ are the output and state of the cell at time step

t. They are generated by several joint computations based on four intermediate vectors: $\boldsymbol{f}_t$, $\boldsymbol{i}_t$, $\widehat{\boldsymbol{C}}_t$, and $\boldsymbol{O}_t$ at time step t. $\boldsymbol{W}_{fx}$, $\boldsymbol{W}_{fh}$, $\boldsymbol{W}_{ix}$, $\boldsymbol{W}_{ih}$, $\boldsymbol{W}_{cx}$, $\boldsymbol{W}_{ch}$, $\boldsymbol{W}_{ox}$, and $\boldsymbol{W}_{oh}$ are trainable weights that appear in pairs for each intermediate vector. $\boldsymbol{b}_f$, $\boldsymbol{b}_i$, $\boldsymbol{b}_c$, and $\boldsymbol{b}_o$ are corresponding trainable biases. As shown in **Table 2.1**, we employed the most recent one time-step electricity load as the only autoregressive feature, and temperature, local time, wind speed, and binary weekday & weekend information ('1' represents weekdays (i.e., Mon-Fri), '0' represents weekends (i.e., Sat or Sun)) as the exogenous features for the prediction. In this case, the exogenous features vector $\boldsymbol{E}_{[t]}$, the input vector $\boldsymbol{I}_{[t]}$, and the predicted hourly load $\hat{y}_{[t]}$ for time step t, are defined as:

$$\boldsymbol{E}_{[t]} = [temperature_{[t]}\ humidity_{[t]}\ time_{[t]}\ windspeed_{[t]}\ weekday\&weekend_{[t]}]$$

$$\boldsymbol{I}_{[t]} = \left[y_{[t-1]}\ \ \boldsymbol{E}_{[t]}\right] \tag{2.4}$$

$$\hat{y}_{[t]} = ConvLSTM\{\boldsymbol{I}_{[t]}\}$$

where $y_{[t-1]}$ is the observed hourly load at time step $t-1$, $\boldsymbol{I}_{[t]}$ is composed by $y_{[t-1]}$ and exogenous vector $\boldsymbol{E}_{[t]}$, as the input vector at time step t, and $\hat{y}_{[t]}$ is the corresponding output (forecasted hourly load). The ConvLSTM cell expects the feature dimension of an individual input to be a two-dimensional array. Therefore, in our case, we take $\boldsymbol{I}_{[t]}$ as a whole, and treat it as one feature with a dimension of one by five.

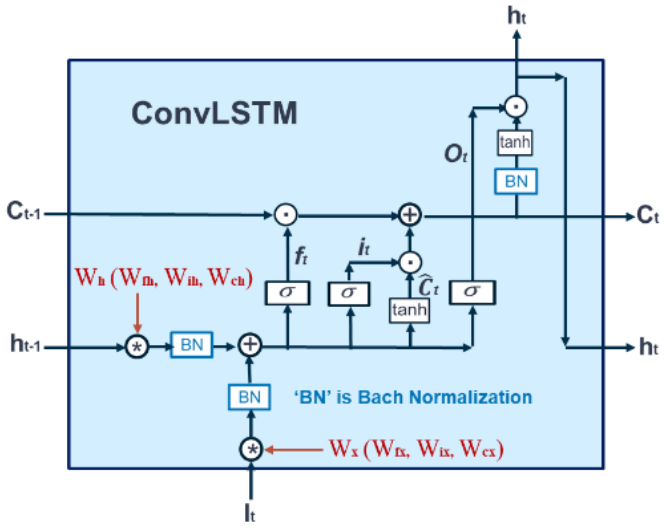

Figure 2.2: Mechanism of a ConvLSTM cell. "•" denotes the Hadamard product and "*" denotes the convolutional operator. Adapted from Xingjian et al. [33] and Marino et al. [16].

As the model is employed to conduct single-step load forecasting, there is no need to separate the data into training vs. testing data. However, the model needs a warm-up period to be initially trained by backpropagation to adjust itself to the best state. Therefore, for each of the three 28-day periods, the first-week load data was used as the warm-up period for initial training, and after that the model was formally employed to make the forecast. After each time-step forecast, the observed hourly load at the last predicted time step was used for parameter updating. Regarding the selection of the previous week as the warm-up period, there are 2 reasons: First, through multiple experiments, we found that when the warm-up period exceeds 2 days, the accuracy of the subsequent forecast will converge. In addition, the forecasting accuracy will not rise if a longer forecasting horizon is implemented, and this is probably due to the varying load patterns as discussed in Section 2.2.1. Second, choosing the previous one week, instead of a longer period, as the warm-up period can make our developed dynamic feature-selection algorithm (to be introduced in Section 2.2.3.3) characterize the historical electricity diurnals of the targeted apartment quickly

without the requirement of significant computational resources. **Table 2.2** shows the hyper parameters of the ConvLSTM model, and **Figure 2.3** shows its un-rolled sequential architecture.

Table 2.2: Structure and hyperparameters of the ConvLSTM model.

Properties	Values
Structure	One ConvLSTM2D layer and two dense layers
Number of filters	36
Kernel size	1x2
Activation function	Relu
Nodes number of first dense layer	4
Nodes number of second dense layer	1
Activation function of dense layers	Relu
Epoch	20
Size of batch	1
Loss function	Mean Squared Error (MSE)
Optimizer	Adam
Training (warm-up) period	First week of each 28-day period
Training time (over CPU)	20 seconds

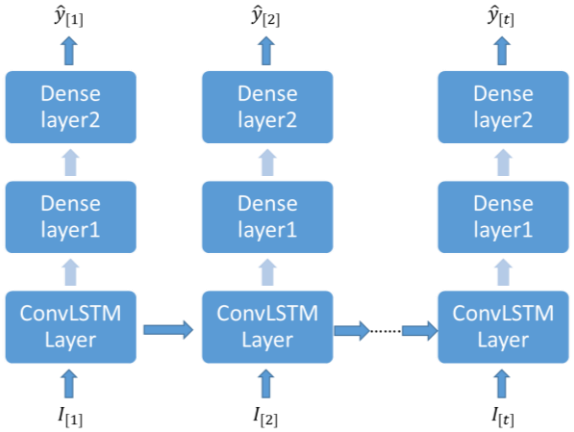

Figure 2.3: The built un-rolled sequential architecture of the ConvLSTM model.

2.2.3.3 ConvLSTM neural network with selected auto-regressive feature (CLSAF model)

As shown in **Figure 2.1**, the load profile in some apartments is characterized by idiosyncratic human behavior (an apartment's temporary vacancy, for example), which could prompt overfitting

in the traditional ConvLSTM model. To preempt such a problem, a better-suited feature representation was devised which selects the most correlated lagged hourly load, rather than locking into inflexibly by using the previous time-step value, or several time-step load values, without any correlation examinations. We thus extended the ConvLSTM model by two additional strategies: An autocorrelation-function (ACF)-based algorithm to select the most correlated lagged load as the autoregressive feature, and a "default" state in which the Persistence model would be employed for prediction whenever the algorithm fails to obtain a lagged load with sufficient correlation as the autoregressive feature. The "default" state, conceived as a model-simplification, is aimed at handling overfitting issues that are mostly caused by load profiles of the same apartment that changed between periods of occupancy vs. vacancy. The CLSAF model was then developed by a combination of the above methods, with the mechanism described in detail below.

First, an autocorrelation function (ACF) was considered for use, which is aimed at selecting the lagged hourly load most correlated with the one-step-ahead load as the autoregressive feature for prediction. The ACF computes the correlation of the time-series lagged values with themselves, thus investigating the periodical nature of a time-series dataset. It is formulated as follows:

$$r_k = \frac{Cov(y_t\ ,\ y_{t+k})}{\sqrt{Var(y_t) \cdot Var(y_{t+k})}} \tag{2.5}$$

where Cov and Var denote covariance and variance, respectively, and y is the observed hourly load at the given time step t or $t+k$. $Var(y_t)$ and $Var(y_{t+k})$ are two variances of the hourly loads with a separation by k time steps. r_k denotes the correlation of the hourly load values with a lag of k hours apart. We set the range of lags returned by the ACF to be 24 ($k = 0, 1, 2, \ \ldots, 24$), to capture diurnal patterns, and r_k is measured over the previous 7 days' hourly load data (starting with the one week warm-up period, which is the longest period available in the dataset prior to

first model employment time step). **Figure 2.4** displays the ACF results of two example apartments.

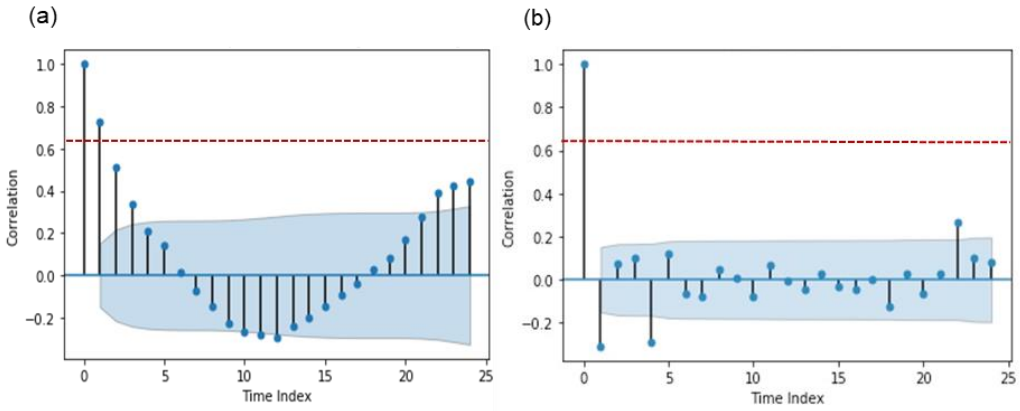

Figure 2.4: Autocorrelation r_k of two example apartments with up to 24 lags (July data).

The ACF is employed in the implementation of the model as follows: First, the ACF is employed to compute the autocorrelation in the previous week's hourly load data to obtain the most correlated lag. Then, the corresponding lagged hourly load is selected as the autoregressive feature, but only for the next one-step-ahead hourly forecast. After moving to the next forecasting time step, the algorithm updates the previous one-week data by adding the latest hourly observation and then repeats the autocorrelation computation and selection process to update the most correlated lagged load for the next-step prediction.

For example, it can be seen in **Figure 2.4** that the first example apartment exhibits a regular pattern with about 24 hours periodicity, and the highest auto-correlation is at the smallest lag considered, i.e., 1 hour. (We ignore the correlation at lag 0 because it is a self-correlated result.) Therefore, in this case, the algorithm would select the load of the previous hour as the autoregressive feature for the next forecast. By comparison, the **Figure 2.4** shows that the 2nd example apartment exhibits only a much weaker diurnal electricity load pattern, and the highest auto-correlation is at lag 23 hours. Therefore, the algorithm in this case would select the load at

lag 23 hours as the feature. However, due to the smaller correlation of the selected lag in the second example, overfitting may occur. This motivated us to improve the CLASF model further by developing a "default" state, as explained below.

Figure 2.5 shows the role of the default state – which acts as a more robust, fall-back option for the load forecasting – and its dynamic implementation in the CLSAF model. We defined a new variable θ, referred to as the autocorrelation threshold, which determines at what time steps the prediction model switches back and forth between the neural network-based forecasting and the forecasting based on the default state. The optimal value of θ was determined by a calibration procedure based on experiments (see next section). As shown in **Figure 2.5**, at every time step, the autocorrelation-based algorithm is employed to select the most correlated lag, as detailed above. Then, the neural network state of the CLSAF model (left path in **Figure 2.5**) is employed for the one-step ahead hourly forecasting by the selected lagged hourly load and the exogenous features stated earlier. However, the resulting model output is used as the CLSAF model's forecast only if the correlation of the selected lag was larger than the threshold θ. Otherwise, the CLSAF model's "default" state (right path in **Figure 2.5**) is activated by using the Persistence model to obtain the forecast for the next time-step. This procedure is repeated at every time step. It is important to note that even after the initial "warm-up" training over the first-week hourly load data, the parameters related to the CLSAF's neural network and the most correlated lag are still updated for each time-step during the forecasting, regardless of whether the actual forecast is taken from the CLSAF's neural network or from its default state. This ensures that the model can switch back seamlessly to the neural network-based forecast whenever the correlation for the most correlated lag is above θ. The exogenous-feature vector $\boldsymbol{E}_{[t]}$, the input vector $\boldsymbol{I}_{[t]}$, and the predicted hourly load $\hat{y}_{[t]}$ of the CLSAF model for time step t are defined as:

$$\boldsymbol{E}_{[t]} = \left[temperature_{[t]}\ \ time_{[t]}\ \ wind_{speed_{[t]}}\ weekday\&weekend_{[t]}\right]$$

$$\boldsymbol{I}_{[t]} = \begin{cases} \left[y_{[t-p_t]}\ \ \boldsymbol{E}_{[t]}\right] & Case\ one \\ y_{[t-1]} & Case\ two \end{cases} \tag{2.6}$$

$$\hat{y}_{[t]} = \begin{cases} ConvLSTM\left(\boldsymbol{I}_{[t]}\right) & Case\ one \\ \boldsymbol{y}_{[t-1]} & Case\ two \end{cases}$$

where the input vector $\boldsymbol{I}_{[t]}$ and the predicted hourly load $\hat{y}_{[t]}$ have two cases. The first case means the load is forecasted by the neural network of the CLSAF model. In case one, $y_{[t-p_t]}$ is the most correlated hourly load selected by the algorithm as the autoregressive feature at time step t. $\boldsymbol{I}_{[t]}$ is the input vector that consists of the selected lagged load $y_{[t-p_t]}$ and the exogenous-feature vector $\boldsymbol{E}_{[t]}$ at time step t, and $\hat{y}_{[t]}$ is the predicted hourly load for time step t. The second case means that the load is forecasted by the CLSAF's default state (i.e., Persistence model). In case two, the predicted hourly load $\hat{y}_{[t]}$ is equal to $y_{[t-1]}$ at time step t.

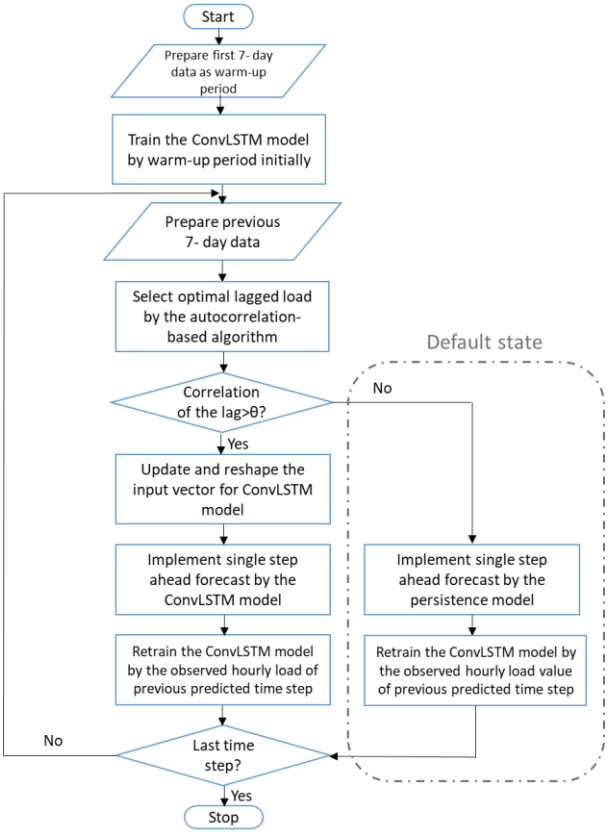

Figure 2.5: The flowchart of CLSAF model.

As shown in **Figure 2.5**, the threshold θ determines which state of the forecasting model is used at which time steps and, therefore, impacts forecasting accuracy. To determine the optimal value of θ, experiments were carried out to measure the average achieved CV-residual at apartment level (sample of randomly selected 20 of the 59 apartments, for 3 seasons). As shown in **Figure 2.6**, the best average forecasting accuracy (i.e., lowest average CV-residual) of the CLSAF model is achieved with θ = 0.64. As θ increases from 0.64 to 0.9, it is increasingly unlikely that the correlation of the selected lag is greater than θ, thus resulting in the default state of the CLSAF being employed more frequently. Similarly, when θ is decreasing from 0.64 to 0.3, it is increasingly likely that the correlation of the selected lag is greater than θ, thus favoring the neural network to produce the load forecast. θ = 0.64 was used in all subsequent analyses.

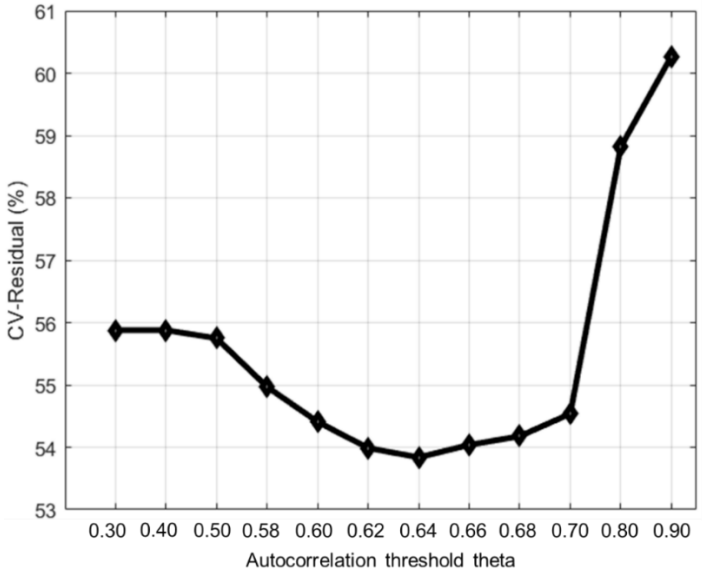

Figure 2.6: Threshold theta vs. average CV-residual of 20 sampled apartments.

2.2.4 Fast Fourier transform (FFT) to assess strength of diurnal pattern

We used frequency spectrum analysis to characterize the daily electricity load profiles, using a Fast Fourier Transform (FFT) algorithm which uses periodicity and symmetry to significantly reduce the computation time [41]. For a sequence of electricity loads y_n at N time steps, the discrete Fourier transform (DFT) is formulated as follows:

$$Y_k = \sum_{n=0}^{N-1} y_n e^{\frac{-2\pi kni}{N}} \tag{2.7}$$

where N denotes the sequence length. To reduce the computational complexity for a more favorable analysis of the spectrum, Eq. (2.7) can be written as:

$$Y_k = \sum_{m=0}^{\frac{N}{2}-1} y_{2m} e^{\frac{-2\pi kmi}{N/2}} + e^{\frac{-2\pi ki}{N}} \sum_{m=0}^{\frac{N}{2}-1} y_{2m+1} e^{\frac{-2\pi kmi}{N/2}} \tag{2.8}$$

where $k = 0, 1, 2, \ldots, N-1$, and $0 \le n < M \equiv N/2$. Y_k is the original amplitude by transformation, in terms of the frequency k.

Once the original amplitudes Y_k for all frequencies ($k = 0, 1, 2, \ldots, N-1$) were determined, we used a scaling approach by standardizing the original amplitudes to generate comparable amplitudes of specific frequencies across the 3 spatial granularities (apartment, floor, and building levels). The standardization was formulated as follows:

$$Y_k^S = \frac{Y_k - \mu_Y}{\sigma_Y} + C \tag{2.9}$$

where Y_k^S is the standardized amplitude of frequency k. μ_Y and σ_Y are the mean and the standard deviation of the original amplitudes. The constant C prevents negative amplitudes and was set to 0.5. In order to quantify the strength of diurnal patterns of the load profiles, we defined a new variable S, as shown in Eq. (2.10), which is the mean value of the standardized amplitudes (Eq. (2.9)) at two specific frequencies, namely, 1 cycle per day, and 2 cycles per day:

$$S = \frac{Y_{k_1}^S + Y_{k_2}^S}{2} \tag{2.10}$$

where $Y_{k_1}^S$ and $Y_{k_2}^S$ are the standardized amplitudes as per Eq. (2.9), and k_1 and k_2 represent the specific frequencies 1 cycle per day and 2 cycles per day, respectively.

2.2.5. Computational resource requirements

The developed ConvLSTM and CLASF models were run on a standard computer with Intel (R) core (TM) 1.99GHz CPU and 16Gb of memory. The code was written in Python. No significant computational resource or code was needed for the Persistence model as the load forecast is simply executed by applying the previous observed hourly load. The ConvLSTM and CLSAF models require approximately the same computational resources because the CLSAF model is a combination of the ConvLSTM and Persistence models. The training (warm-up) period for each lasted about 20 seconds, and only 0.2 seconds were required for each subsequent time-

step for parameter updating and prediction. Therefore, a standard machine with one CPU could easily provide the required computational power for a real-life application of the CLASF model in a TEN, meaning that each next hour load could be forecasted near instantaneously as soon as the previous time step's load has been measured and exogenous variables have been collected.

2.3 Results

2.3.1 The best performing models in the study

As previously discussed, the principal challenge of load forecasting with respect to our dataset is the large volatility of loads in individual apartments. Thus, in selecting the best performing models, our priority was focused on the performance of all models in forecasting apartment-level load data. An overall summary of apartment-level forecasting accuracies by the 6 employed models (4 benchmark models and two newly employed models) is provided in **Table 2.3**.

Table 2.3: Overall average CV-residuals of apartment-level load forecasting for all three datasets (January, April, and July) by the 4 benchmark models and the 2 newly employed models.

Model name	Mean value	Minimum	Maximum
Persistence	61.2	6.3	141.4
SW-ARIMA	64.1	6.4	201.5
SW-ETS	63.7	6.4	188.4
SW-SVR	62.0	6.3	162.3
ConvLSTM	**57.9**	**6.2**	**131.1**
CLSAF	**53.3**	**5.9**	**115.8**

As shown in **Table 2.3**, the SW-ARIMA, SW-ETS, and SW-SVR models have worse accuracy (higher CV-residuals) than the ConvLSTM and the CLSAF models when handling the case of individual-apartment load forecasting. Notably, their accuracies are even lower than the accuracy of the persistence model. Therefore, in the following, we mainly pay attention to the forecasting results of the persistence model, the ConvLSTM model, and the CLSAF model.

2.3.2 CV-residuals by spatial granularity, models, and seasons

The forecasting accuracies of the three models (the persistence, ConvLSTM, and CLSAF models), evaluated by CV-residual (Eq. (2.1)), for all scenarios (three spatial granularities and three seasons) are provided in **Figure 2.7**. The forecasting accuracy varies as a function of spatial granularity, model type, and season, as analyzed in more detail in the following sections.

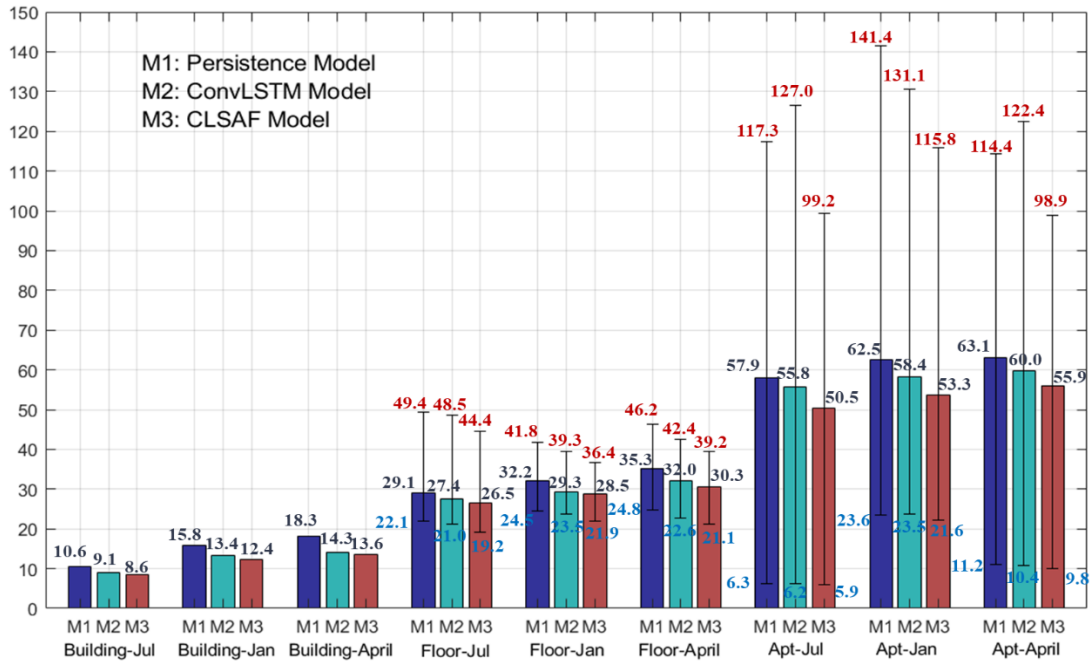

Figure 2.7: Forecasting accuracy (CV-residual, in %) of building, floor, and apartment level over the 3 seasons. Error bars indicate the maximum and minimum CV-residuals of each group (red and blue numbers, respectively); black numbers give the averages. The building level has only one forecast accuracy for each model and season.

2.3.2.1 Effect of spatial granularity on forecasting accuracy

Reviewing the results in **Figure 2.7**, it is easy to note that the highest average accuracy is achieved at the building level (lowest CV-residual), followed by floor level and then apartment level (highest CV-residual). To test this result for statistical significance, we carried out two-sample t-tests (two-tailed, unequal variances).

For the floor vs. apartment level, nine such tests (Floor-Jan & Apt-Jan, Floor-April & Apt-April and Floor-Jul & Apt-Jul across three models) were carried out. These showed that the average accuracies of all 9 combinations are significantly different ($p<0.05$), confirming that the floor level forecasting outperforms that at apartment level.

For the building level, no further statistical tests were carried out because our dataset only contained one building. However, as seen in **Figure 2.7**, the CV-residual at the building is smaller than even the minimum CV-residual of any of the floors. Consequently, the building level produces the highest forecasting accuracy for our dataset.

2.3.2.2 Effect of model type on forecasting accuracy

As shown in **Figure 2.7**, the CLSAF model yields the highest average accuracy, followed by the ConvLSTM model, and then the Persistence model. To verify the statistical significance of this finding for floor and apartment levels, nine paired t-tests (two-tailed, unequal variances) were carried out (Persistence & ConvLSTM, Persistence & CLSAF and ConvLSTM & CLSAF, across three seasons). The results show that the averages of all 9 combinations are significantly different ($p<0.05$).

One typical example that illustrates an advantage of the CLSAF model compared to the ConvLSTM model is shown in **Figure 2.8**: When either model is confronted with a period of vacancy in an apartment, the CLSAF model reacts to the change faster, regardless of whether the apartment changes from occupied to vacant (around August 2nd in **Figure 2.8**) or vice versa (after August 10th). This is because the CLSAF model can switch its state back and forth between the neural network and the Persistence model, thus mitigating overfitting due to volatile load data, as stated earlier. In contrast, the load forecasted by the ConvLSTM model shows a continuing diurnal

variation for the full period of the vacancy, because it overfits to the pre-vacancy period, thus leading to smaller forecasting accuracy.

Furthermore, as can be seen in **Figure 2.9 (a)**, the pattern of average forecasting accuracy shown in **Figure 2.7** does not hold for all apartments individually. While the average CV-residual of the ConvLSTM model for all three seasons are lower than the ones of the Persistence model, the situation is reversed for some apartments (e.g., red circle in **Figure 2.9 (a)**). The reason is that the ConvLSTM model sometimes loses robustness leading to overfitting, as illustrated in the example apartment in **Figure 2.8** (which shows the observed and forecasted load profiles of the same apartment as the one highlighted by the red circle in **Figure 2.9 (a)**). Overall, such possible overfitting is avoided by the CLSAF model which outperforms both the Persistence model and the ConvLSTM model, not only on average, but for every apartment, floor, building, and season, individually.

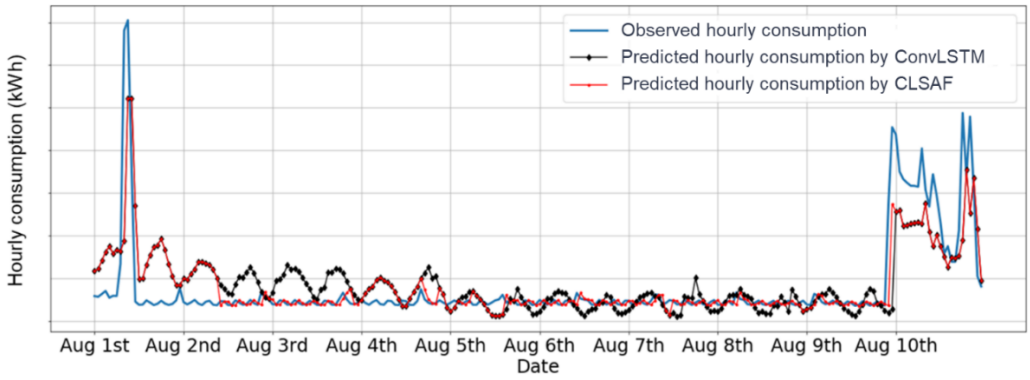

Figure 2.8: Hourly forecasting results of the ConvLSTM and the CLSAF models for one example apartment from Aug 1st to Aug 10th, 2019. Exact hourly load values are not shown for privacy considerations.

2.3.2.3 Effect of season on forecasting accuracy

In addition to the above-mentioned effects, it can be noticed that, for a particular spatial granularity and model type, forecasting accuracies are considerably affected by the season, with

July consistently exhibiting the lowest (i.e., best) average CV-residual, followed by January, and then April. We again used two-sample t-tests (two tailed, unequal variances) to determine whether the exhibited differences in forecasting accuracy caused by seasonal changes are statistically significant. Nine pairs were set up for the floor or building level (January & April, January & July and April & July, across three models). The results show that the differences in the average CV-residual of apartments [of floors] between the different seasons are not statistically significant ($p > 0.05$), owing to the large variation in each sample of 59 apartments [11 floors] and the limited sample sizes. Consistent with that, intra-group variance of CV-residual, determined via ANOVA, is substantially larger than inter-group variance, as evidenced by $(1 - \eta^2) = 0.98$ for apartments [0.88 for floors].

Such high level of intragroup variance in CV-residual – which is not explained by the spatial granularity or model type – points to the possible existence of other not yet identified characteristics in each observed load profile. This will be explored in the next sections.

2.3.3 Volatility of electricity consumption vs. forecasting accuracy

The large variations in CV-residual in **Figure 2.7** and **Figure 2.9 (a)** indicate different levels of forecasting-"difficulty" for different apartments and/or floors. Therefore, we searched for underlying characteristic of the electric load profiles that impacted forecasting accuracy. One such characteristic was found to be the volatility of the load data, henceforth CV-observation. The definition of CV-observation is as follows:

$$CV_{observation} = \frac{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2}}{\bar{y}} \tag{2.11}$$

where y_i is the observed hourly load at the ith time step. $\bar{y}$ denotes the mean value of the observed loads, and N is the number of the observations (here 672, for hourly data over 28 days). CV-observation can be understood as a type of normalized standard deviation and reflects a scaled variation of electricity load. Therefore, a load profile with a larger mean value but similar absolute standard deviation has a smaller CV-observation. **Figure 2.9** shows the relationship between CV-observation and forecasting accuracy, along with the respective linear correlations and p-values. July was randomly chosen as the example to visualize the relationship. For January and April, results are similar to those in July, namely all correlations are between 0.62 and 0.69, and all p-values are smaller than 5e-6. This shows that, regardless of model type, season, or spatial granularity, the achieved forecasting accuracy is driven to a considerable extent by CV-observation of the load profile, with prediction accuracy the higher, the lower CV-observation.

The relationship between average CV-observation and average CV-residual across the three seasons is shown in **Figure 2.9 (b)**. July data yields the lowest averages of the two metrics, and April data the highest. Since this is consistent with the pattern in **Figure 2.7**, this provides a likely explanation for why the three seasons exhibit different average CV-residuals: As the average CV-observation increases from July to January and April, the average achievable forecasting accuracy decreases accordingly. In other words, the seasonal effect on forecasting accuracy is at least partially explained by a concurrent difference in CV-observation between the seasons.

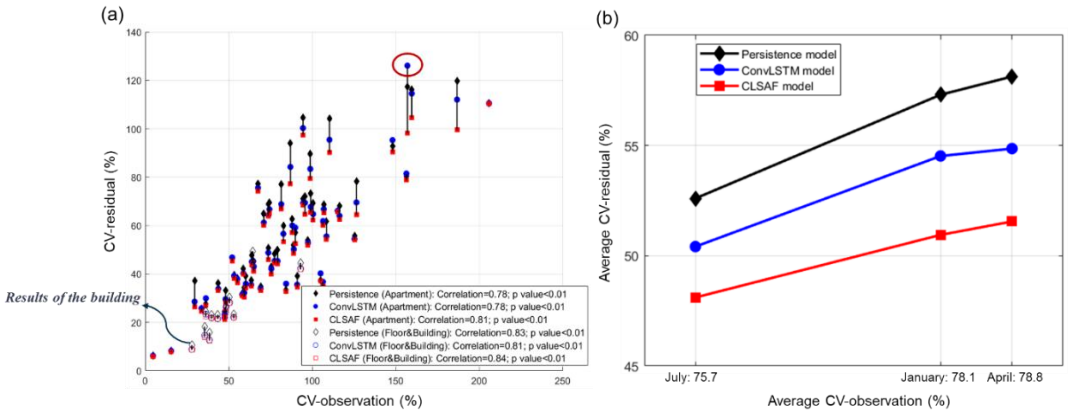

Figure 2.9: (a) CV-observation vs. CV-residual, for three models and three spatial granularities (July data). (b) Average CV-observation by three seasons vs. respective average CV-residual.

2.3.4 Strength of diurnal patterns vs. forecast improvement

Examining **Figure 2.9 (a)** shows another effect that, however, does not seem to be easily explained by CV-observation: The improvement in forecasting accuracy from the Persistence model (benchmark) to the CLSAF model varies between apartments (as well as between floors). This led us to search for a characteristic of the load profiles that affected this accuracy improvement. As illustrated in Section 2.2.3, the key difference between the Persistence model and the CLSAF model is that the latter employs a feature-selection techniques that can extract the core information of daily load profiles. Therefore, the difference in accuracy between these two models is likely due to how much of such daily-profile information is present in a particular profile.

In order to investigate the extent to which the daily profiles aided higher forecasting accuracy of the CLSAF model vs. the Persistence model, we defined a new variable *S* to quantify the strength of diurnal patterns of the load profile, as defined in Methods. To illustrate graphically which load characteristic *S* is sensitive to, **Figure 2.10** shows the spectral analysis of two load profiles, one with strong diurnal periodicity at the 12h and 24h mark, and one without.

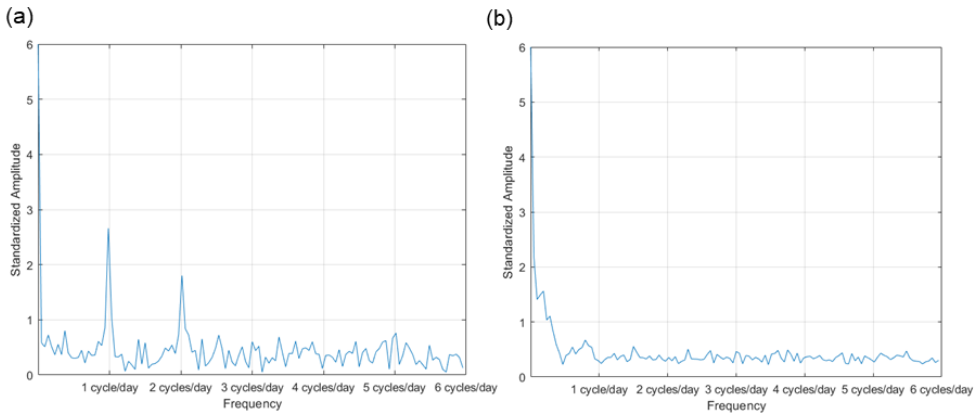

Figure 2.10: **Spectral analysis of two sample apartments by FFT with the standardized amplitude. The spectrum in (a) reflects a strong diurnal pattern, evidenced in the spikes at 1 and 2 cycles per day, respectively. The spectrum in (b) reflects few to none diurnal electricity patterns.**

The improvement in forecasting accuracy, i.e., reduction in CV-residual, R was defined as follows:

$$R = \frac{CV_{residual\ (persistence)} - CV_{residual\ (CLSAF)}}{CV_{residual\ (persistence)}} \tag{2.12}$$

where CV-residual is as defined in equation (1). **Figure 2.11** shows the relationship between the strength of diurnal patterns (S) and the forecasting accuracy improvement (R) for all spatial granularities and seasons.

The results demonstrate that the strength of the diurnal pattern has a statistically significant impact on the forecasting accuracy improvement, with an improvement of up 25% in some cases. The apartment-level has the smallest average improvement (R = 11%), followed by floors (R = 14%), and buildings (R = 23%). The result underlines that, as outlined in Section 2.3.2, it is inherently more difficult to predict electricity load profiles whose diurnal profiles are either not present or masked by high volatility. In contrast, a stronger diurnal pattern, which tends to me more pronounced in the aggregated loads of an entire floor or building, facilitates the information extraction and learning process executed by more complex models such as the CLASF model, thus resulting in larger forecasting accuracy improvement for such models vs. the benchmark Persistence model.

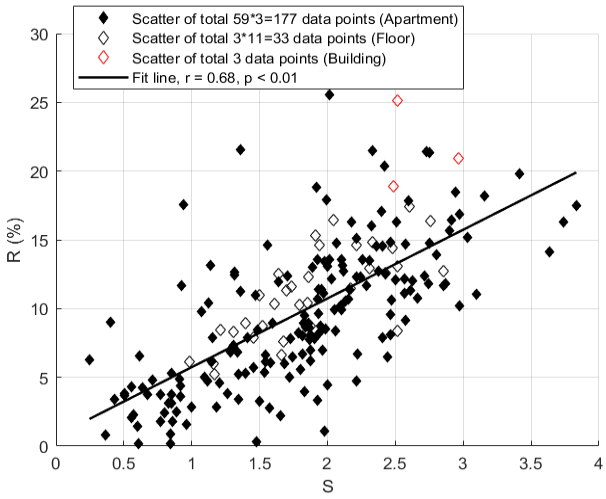

Figure 2.11: **Strength of diurnal pattern (S) vs. relative reduction of CV-residual (R) from Persistence model to CLSAF model, covering the results of 3 spatial granularities and all 3 seasons.**

2.3.5 Combination of CV-observation & diurnal pattern strength vs. CV-residual

Next, we sought to understand to what extent the above two underlying characteristics (CV-observation and strength of diurnal patterns) in combination can explain the achieved forecasting accuracy. This is shown in **Figure 2.12**, which divides the parameter space of CV-observation and *S* into four areas representing four load profile categories, using the averages of CV-observation and *S* as the area separation points. We classified all 213 CV-residuals obtained by the CLSAF model (59 apartments, 11 floors, and 1 building; each for 3 seasons) into the four categories according to their corresponding CV-observation and S.

We found that electricity load profiles with high *S* and low CV-observation yield the highest average forecasting accuracy (i.e., lowest CV-residual). The opposite is true for load profiles with low *S* and high CV-observation. Furthermore, the effect of CV-observation on forecasting accuracy is stronger than that of *S*, as seen by changes of 26 percentage points in CV-residual along the CV-observation dimension but only 8 percentage points in the *S* dimension. To test for statistical significance of these effects, we carried out two-sample t-tests (two-tailed, unequal

variances). We found statistical significance ($p < 0.01$) for 5 of the 6 pair-wise differences, and moderate statistical significance ($p = 0.06$) for one pair-wise difference (**Table 2.4**).

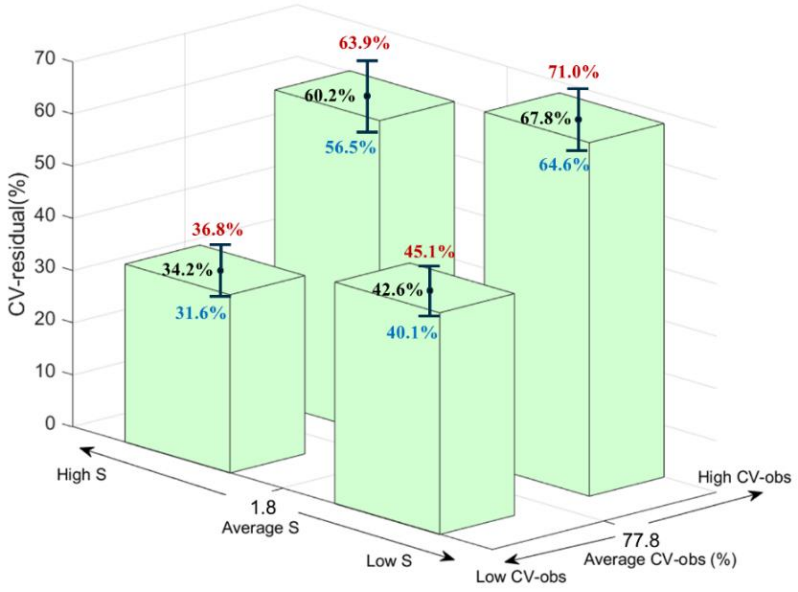

Figure 2.12: CV-observation vs. strength of diurnal pattern (*S*) and average CV-residual for 4 load profile categories.

Table 2.4: Results of t-tests (two-tailed, unequal variance) to determine statistical significance of the differences in average CV-residual per load profile category.

p values for corresponding t-test	High CV-observation & high S	High CV-observation & low S	Low CV-observation & high S	Low CV-observation & low S
High CV-observation & high S	N/A	p=0.06	p=9e-12	p=3e-21
High CV-observation & low S		N/A	p=1e-5	p=5e-10
Low CV-observation & high S			N/A	p=2e-3
Low CV-observation & low S				N/A

2.4 Discussion

Our results could serve as a starting point to set up a possible data pre-assessment method for time-series electricity-load datasets. The method would allow users of load forecasting models to make a preliminary assessment of the nature of a load profile dataset, providing two benefits: (i)

reducing the modeling complexity for some apartments; and (ii) providing confidence levels for the predicted electricity use.

With regards to the first benefit, possible implementation steps would be as follows: First, one could use the previous 28-day electricity data of an apartment intended for forecasting to compute CV-observation. Using the relationship described in **Figure 2.9**, this would provide an approximation for the forecasting accuracy likely achievable by even a simple Persistence model. Second, one could use the spectral analysis to determine the load profile's strength of diurnal pattern *S*, again using the previous 28-day data. Using the relationship illustrated in **Figure 2.12**, CV-observation and *S* together would provide an estimate of the forecasting accuracy of the CLASF model.

As for the 2nd benefit, knowing not only the forecasted electricity use, for example for the next hour, but also the confidence levels of the prediction (inferred from CV-residual) would allow more sophisticated transaction schemes within the examples of TEN applications outlined in Introduction, as follows: Any such trading of electricity with others would carry risks – namely the risk of either not having enough electricity for one's own use or, alternatively, not being able to honor the transaction agreed to with another user. However, the ability to evaluate how accurate the forecast will likely be, makes these risks more manageable. For example, user A may be able to determine that despite having committed to selling a certain number of kWh from their own storage to user B, user A can still be 90% confident to have enough electricity for themselves. Alternatively, the transaction could be priced such that user B knows that there is a 10% risk that user A will not be able to provide the full amount of electricity that they agreed on.

2.5 Conclusions

In this chapter, first, we present a novel ConvLSTM neural network model with selected autoregressive features (CLSAF model) to improve single-step-ahead electricity load forecasting for three spatial granularities: apartment, floor, and building level. The CLSAF model achieves higher forecasting accuracy (up to 25% improvement vs. the Persistence model). The CLSAF model enables durable robustness by leveraging the advantages of its autocorrelation-based feature-selection algorithm and a model-simplification method to prevent overfitting when confronted with volatile load data caused by changes in resident behavior and/or temporary absences.

Second, based on the prediction results of our multi-granularity dataset across the three seasons, we present a load-profile-identification strategy for two characteristics that are statistically significantly correlated with forecasting accuracy, namely CV-observation and the strength of the diurnal pattern S. These characteristics capture the load profile volatility and the degree of learnable daily-profile information, respectively. The smaller CV-observation and the stronger the diurnal pattern, the higher is the forecasting accuracy the CLSAF model can achieve. Moreover, we discuss how these conclusions can guide basic steps of a possible data pre-assessment method for practical load forecasting applications and the associated advantages.

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE), under the Building Technologies Office’s, BENEFIT program, Award Number DE-EE-0007864.

Chapter 3. COVID-19 Related Impact on Residential Load and Grid Stability

The main part of this chapter is presented in the paper co-authored with Prof. Christoph Meinrenken, Prof. Vijay Modi and Prof. Patricia Culligan, and published in the Journal of Energy and Buildings [42].

3.1 Introduction

3.1.1 Background and motivation

Since early 2020, the COVID-19 pandemic has caused a global catastrophe, impacting almost every aspect of daily life in most countries. In early 2020, approximately one third of the world's population was in "lockdown" via various types of "stay-at-home" orders or similar guidelines. This severe situation saw more than 80% of workplaces worldwide partially or fully closed, resulting in significant economic impacts, including a global recession that might rival the Great Depression [43].

Generally, how to effectively respond to global disasters is a crucial issue for local governments and decision-making personnel. Energy and electricity infrastructures (from energy supply to demand) have faced disruptions due to the COVID-19 pandemic and related shelter-in-place orders that are believed to be the most severe in seven decades [44]. Worldwide, the partial or complete shutdown of many commercial and social activities has substantially reduced energy demand in 2020 [43]. To investigate the changes in electricity profiles due to the pandemic, Bahmanyar et al. [45] compared the effect of different containment policies carried out by 6 European countries (Spain, Italy, Belgium, the Netherlands, Sweden, and the UK) on their

electricity consumption during the COVID-19 pandemic. They found that the weekday consumption of most of them considerably decreased and that the consumption profiles were close to pre-pandemic weekend profiles when compared to the same period in 2019.

Although the overall energy consumption during the pandemic decreased, the decrease was driven by reduced commercial loads in large metropolitan areas such as NYC, London, or Paris, whereas residential electricity consumption increased as many residents switched to working or undertaking educational or other activities from home [46]. In addition, the shape of the residential energy demand profile shifted, with weekday diurnal profiles resembling pre-COVID-19 weekend diurnals [47]. Some studies showed electricity peaks disappearing during morning periods, with these peaks instead shifting to noon. For example, one study reported an approximate 30% increase in electricity use around midday in the U.K. during early April 2020, compared to pre-pandemic times [48]. In addition, in the NYC metropolitan area, also in early April 2020, a 23% increase during typical working hours (9:00 am to 5:00 pm) was observed [49].

Significant changes in household day-time use would lead to new load profiles that might produce new challenges for the consumers and for the grid. In early 2020, many settings of utilities and governments allowed customers to defer payments, leading to large past-due electricity bills, and the electricity bills in the summer months have also been higher than pre-pandemic bills [50]. Even in heating dominated-geographies such as New York City, one experiences hot weather, and during those periods the cooling demand can dominate. This need is met through the use of electricity, unlike much of the heating. Hence one would expect that if residents spend more time at home between 9 am and 5 pm on weekdays than they would have otherwise, the energy use during that period would be higher. One way to reduce residential summer peak load is to incentivize behavioral modification, e.g., encouraging residents to curb on-peak electricity-usage,

such as for laundry, by shifting respective activities to other times of the day [51]. For managing summer peaks during global crises, such as the COVID-19 pandemic – or even national-level crises, such as the 2011 Japan Earthquake – more factors need to be taken into account, including how hot it gets over the summer months, and whether more residents are allowed, willing or even encouraged to return to the usual place of work/school during the aftermath of a crisis [50].

3.1.2 Focus and objective of this chapter

A case study is conducted to investigate Covid-19-related increases in residential electricity usage from 2019 to 2020, based on the electricity data recorded in the same database of multi-family residential buildings considered in Chapter 2. For the analysis below, we focus on two characteristics of the electricity usage of an average apartment, (i) the electricity consumption (kWh) on weekdays during the 8 hours from 9am to 5pm (in order to gauge how much electricity use and commensurate financial burden shifts from commercial buildings and schools to the residential sector); and (ii) the hourly peak demand (Watt) on weekdays during the 5 hours between 12pm and 5pm (in order to gauge possible stress on the electricity grid when increased residential peak demand either coincides with system-wide loads or becomes larger than the substations and/ or distribution lines in residential areas were designed to handle). We develop a series of robust predictive models and identify two key drivers of residential electricity usage, namely the severity of the pandemic – as measured by the Covid-19 case load – and the outdoor wet-bulb temperature. By performing the Monte Carlo simulation, we then use these models to predict electricity usage characteristics for conditions when there is a confluence of high outdoor temperatures during the summer with medium to high portions of residents working or studying from home. Such conditions might occur if COVID-19 stay-at-home orders in urban areas like NYC persist into the

summer months – or if there is widespread adoption of a work and study from home lifestyle that is non-pandemic related but part of a future, "new normal". The predictions are used to understand how much residential summer electricity peaks might increase financial burdens for residents and the risks of grid stress or failure.

3.2 Data and methods

3.2.1 Dataset for apartment-level electricity usage

3.2.1.1 Electricity related to heating and cooling

In this work, the same electricity database of MFRED considered in Chapter 2 is used, which covers 390 apartments ranging in size from studios to 4-bedroom units [37]. The heating in 89% of the apartments is supplied centrally (burning natural gas and distributed within building, using steam or hot water), whereas the air conditioning is supplied by personal appliances that are commonly the window air conditioners mounted on windows or walls using electricity. Therefore, heating in most apartments does not contribute to the apartments' own electricity usage (except for heating blankets or space heaters) but air conditioning does. The other 11% of apartments are equipped with different forms of packaged terminal air conditioners (PTACs), with the majority of the cooling and heating supplied centrally, such that the PTACs' electric load does not materially contribute to an apartment's electricity usage. Therefore, the vast majority of apartments in our dataset exhibit higher electricity use during the summer, depending on weather conditions, especially temperature. In contrast, the electricity usage during the winter and shoulder seasons depends much less on the weather. For analysis, we used the incremental electricity consumption (kWh) from one hour to the next from January 1st to August 31st of both 2019 and 2020. The 2019 and 2020 data were compared to reveal modified diurnal shapes and increases in both consumption

(kWh) and peak demand (Watt) due to the effects of stay-at-home conditions during the pandemic in 2020.

3.2.1.2 Removing vacant apartments from dataset

Before analyzing the overall daily electricity use of apartments, we sought to eliminate the impact of uninhabited apartments on average electricity consumption. Therefore, apartments that were not occupied for a long period of time (henceforth "vacant apartments") were removed from the dataset.

To robustly identify vacant apartments, a threshold *T* for the 1-month average load of an individual apartment was set at 1.067 Watts per square meter (W/m^2). The value was determined for the average size of studios and one-bedroom apartments of our dataset, which usually have a minimum consumption of 70 Watt that consists of a refrigerator (~50 Watt) plus ~20 Watt for a router/Wi-Fi and other electronics in standby mode. The electrical consumption of refrigerators in the vacant apartments can vary considerably with the changes of climate conditions. Therefore, such a definition of the threshold does not consider the additional load caused by weather condition changes, and the 1.067 W/m^2 threshold thereby should be only applicable in the shoulder seasons, and thus was used for April only. To determine the thresholds suitable for identifying vacant apartments in other months, the April value was scaled in proportion to the typical electricity consumption of all 390 apartments in the respective month, as follows:

$$T_{month} = T_{April} \times \frac{B_{month}}{B_{April}} \tag{3.1}$$

where T_{April} is the April threshold (1.067 W/m^2), T_{month} is the threshold of any month, and B_{April} and B_{month} are the baseline consumptions, defined as the time-averaged apartment electricity load during April and the targeted month, respectively.

By the defined threshold, the numbers of identified temporarily vacant apartments from Jan. to Aug. in 2019 and 2020 are computed and shown separately in **Figure 3.1**. One can easily observe that an increase in the number occurs after February 2020, probably due to the outbreak of the pandemic in NYC, which prompted some residents to move out of their apartments temporarily. In order to maximize consistency between the 2019 and 2020 datasets (i.e., same apartments in both years), an apartment was removed from both datasets whether it was deemed vacant in 2019, in 2020, or both. Based on this approach, 84 vacant apartments were removed from the 2019 and 2020 data, leaving 306 apartments for all subsequent analyses.

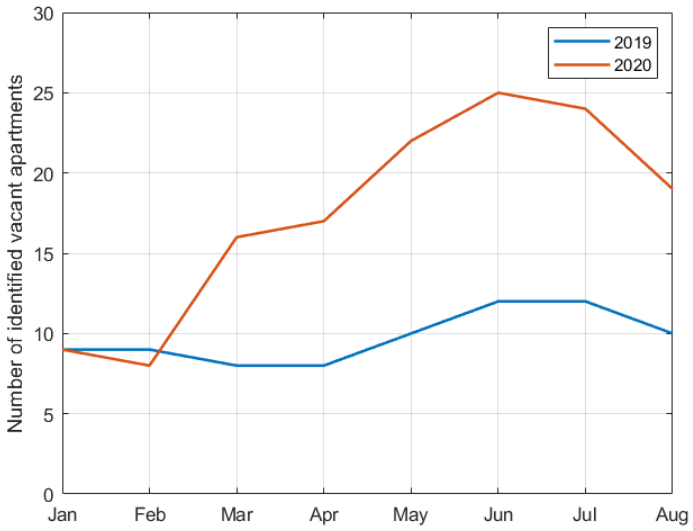

Figure 3.1: Numbers of identified vacant apartments from Jan to Aug in 2019 and 2020.

3.2.1.3 Electricity consumption baseline adjustment for 2020 data

Electricity consumption in the 306 apartments might have changed from 2019 to 2020 for reasons other than the pandemic. This effect was accounted for via a baseline adjustment. Since the residents' work and study patterns started changing in NYC only from March 2020 onwards, the electricity data from Jan. 1 – Feb. 29, 2020 was not yet impacted by the pandemic. Therefore, this period was chosen as a benchmark to reveal any difference in electricity-use baselines between

2019 and 2020. The average usage in Jan. – Feb. 2020 was 2.0% lower than during Jan – Feb. 2019. One possible reason could be the adoption of more energy-efficient devices such as LED light bulbs or electronics with lower stand-by power consumption. To show that the difference of the electricity-use baseline between 2019 and 2020 is not due to weather conditions, especially the temperature that is the key factor impacting electricity demand, we investigated the average monthly electricity consumption and the average daily wet-bulb temperature in Jan. and Feb. of 2018, 2019 and 2020, These are shown in **Table 3.1** for each year. One can observe that although the average temperature in Jan. – Feb. of 2019 is 0.2^{o}C and 2.4^{o}C lower than the ones in 2018 and 2020, respectively, it is the monthly electricity consumption in Jan. – Feb. of 2018 that stands the highest, about 2.2% larger than the one in 2019, indicating that the weather condition is probably not the key factor leading to the decrease of the electricity-consumption baseline in Jan. and Feb. of 2020. Therefore, to show the difference between the electricity diurnals and use of the two years more accurately, the hour-to-hour electricity consumption for 2020 was increased by 2.0%. All subsequent analyses, results, and figures in this chapter reflect the 2020 data after this adjustment.

Table 3.1: The monthly electricity consumption and average daily wet-bulb temperature in Jan. and Feb. of 2018, 2019, and 2020.

	January		February	
Years	Monthly electricity consumption (kWh)	Average daily temperature (^{o}C)	Monthly electricity consumption (kWh)	Average daily temperature (^{o}C)
2018	260.91	-2.1	231.21	3.5
2019	254.30	-2.3	226.92	-0.1
2020	249.65	0.1	221.89	2.5

3.2.2 Choice of relevant factors and time-windows of interest

3.2.2.1 Preliminary analysis of factors driving residential electricity usage patterns

In order to analyze in what time-windows the residential electricity usage has changed most significantly due to the pandemic in 2020, an electricity-diurnal analysis was carried out. For brevity, we henceforth refer to the times before March 21st, 2020, as the "pre-stay-at-home" period, and the times after that as the "stay-at-home" period.

First, it can be noted from **Figure 3.2 (a)** that there are shifts in demand during the morning hours on weekdays: During the pre-stay-at-home period, the early-morning load ramp-up started at about 6.00am and peaked at 8.30am, followed by a decline, with no second ramp-up until the early evening. In contrast, stay-at-home usage exhibited a smoother ramp-up that started between 6.00am and 6.30am, reached the height of the pre-stay-at-home morning demand peak only at 9.00am, and then continued to increase through the morning and early afternoon.

Regarding electricity use, **Figure 3.2 (a)** shows that, overall, 2020 weekday electricity usage of apartments (24h) shows a more significant increase (7% increase) versus 2019 use than on weekends (4% increase). These increases became more pronounced once advancing into warmer weather in July, where the increase in 24h weekday-use above 2019 reached 13%, probably due to higher loads from air-conditioners (**Figure 3.2 (b)**).

Studies for commercial buildings in the U.S. have shown that their principal electricity use is mostly concentrated in the worktime period (usually 9 am – 5 pm) on weekdays [52]. Focusing on the same time window in the residential sector, when many residents would usually be at work/school or otherwise outside of their homes, the stay-at-home usage increases are even larger than over the 24h period: Comparing 2020 to 2019 usage during 9 am to 5 pm, one can see a 22% increase in average electricity use in early April and an even larger increase of 27% in early July.

Figure 3.3 shows the overall trends in the 24-hour-electricity-use and 8-hour-electricity-use (9 am—5 pm) as percentage increases from 2019 to 2020, over the same period of Jan 1st – Aug 31st. Percentage increases in the hourly peak demand on weekdays between 12pm and 5pm are also shown (see rationale in Section 3.2.2.3). It can be observed that the three characteristics, especially the hourly peak demand between 12pm and 5pm and the 8-hour electricity use, are correlated with two metrics, i.e., the outdoor wet-bulb temperature and the number of new confirmed Covid-19 cases in every month: During the stay-at-home period, the pandemic led to significant increases in residential electricity use, even when temperatures had not yet reached levels where air conditioning was required. These increases were therefore most likely due to an increased use of lights, appliances for food preparation, computers, and entertainment systems because more residents worked/studied from home. Once entering Phase 1 of the gradual reopening, new daily Covid-19 cases in NYC were declining, and the portion of residents remaining in their homes during the day was likely declining as well [43]. However, due to the higher outdoor temperatures now requiring increased cooling loads, the 8-hour electricity usage exhibits a notable further increase during the summertime in 2020.

In summary, both the outdoor temperature and Covid-19 cases should be considered when explaining differences in electricity usage between 2019 and 2020.

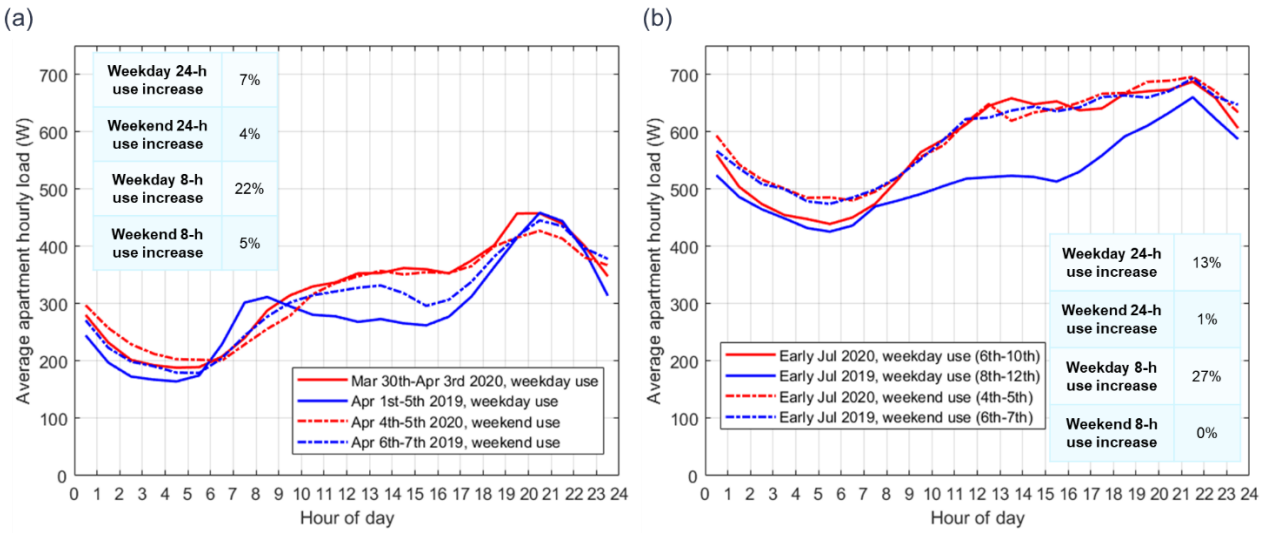

Figure 3.2: (a) Stay-at-home and pre-stay-at-home electricity diurnals of one week in early April of 2019 and 2020, respectively. (b) Same for one week in July.

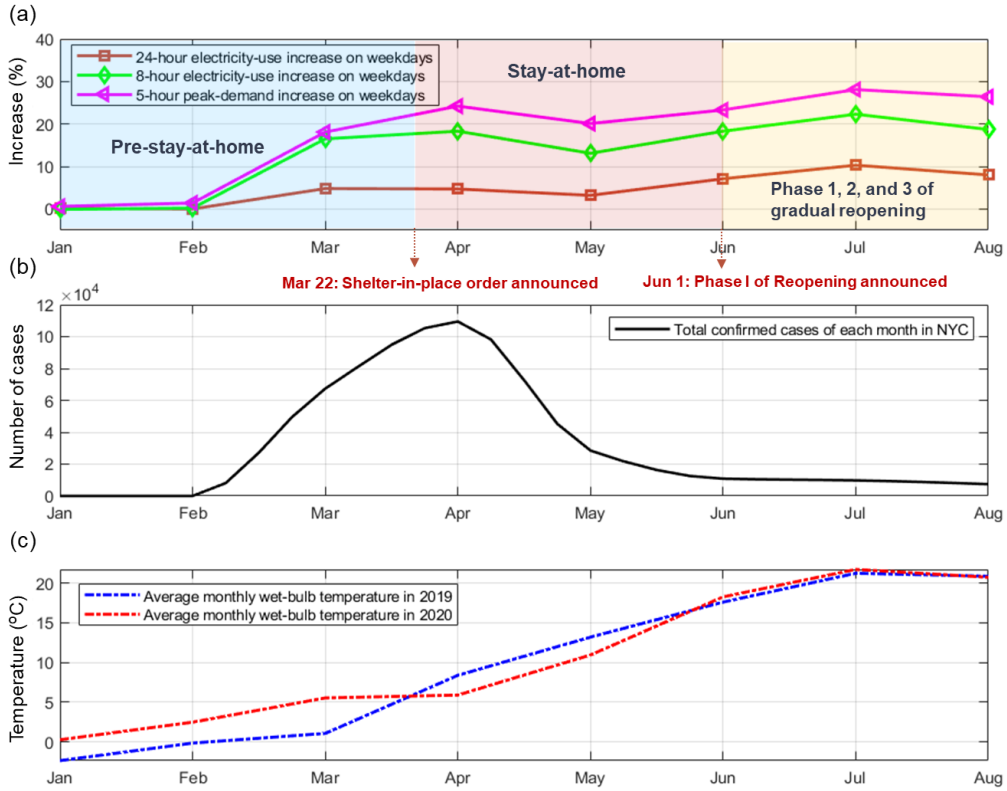

Figure 3.3: (a) Increases in 24-hour-use, 8-hour-use, and 5-hour peak-demand (weekdays) between 2019 and 2020, by month. (b) Total monthly new confirmed Covid-19 cases in NYC in 2020, by month. (c) Average monthly wet-bulb temperature in 2019 and 2020, by month.

3.2.2.2 Choice and rationale for time windows and electricity metrics of interest

Based on the observations in Section 3.2.2.1, for the remaining analyses, we therefore focus on the following two characteristics of electricity usage, which capture different time windows and different electricity metrics:

(i) Average per-apartment electricity consumption (kWh) cumulatively from 9am to 5pm on a given weekday, for brevity also referred to as "8-hour-electricity-use". This was analyzed in order to gauge the electricity usage (and associated costs) that can shift from the commercial sector (such as office buildings and schools) to the residential sector because of "stay-at-home" and/or "work-from-home" guidelines.

(ii) Hourly peak demand (Watt) for an average apartment at any time between 12pm to 5pm on a given weekday, defined at 1-hour resolution, for brevity also referred to as "5-hour-peak-demand". "Peak demand" was defined as the highest of the hourly average load (in Watts) between any two consecutive full hours in the time window of interest. To establish these, first, the hourly average Watts between 12-1pm, 1-2pm, … and 4-5pm on a given day were determined, and then the "5-hour-peak-demand" on that day was taken to be the maximum of these five, hourly values. The peak demand during full or partial stay-at-home orders was further compared to the highest ever hourly residential peak in a no-pandemic condition in 2019. This peak typically occurs in the evenings of hot/humid days. The comparison was carried out in order to gauge whether the increased afternoon peak demand during widespread stay-at-home conditions could lead to black-outs or brown-outs of the local substations and distribution system in predominantly residential regions of a city (because the demand is larger than what the system was designed to handle).

3.2.3 Model components and calibration

3.2.3.1 Model inputs and outputs

Previous work on electricity usage forecasting for households has shown that outside temperature is the strongest factor driving electricity demand in the residential sector, if the cooling systems of the targeted households, as in our case study, comprise electrical air conditioners [51]. Regarding the specific type of temperature, previous work has shown that wet-bulb temperature is a better predictor for residential cooling loads than dry-bulb temperature, as the former captures both temperature and humidity [21]. Therefore, we chose wet-bulb temperature (henceforth *WBT*) as our first independent factor for modeling. *WBT* was available at approximately hourly time resolution, typically with a data point available near the full hour (National Oceanic and Atmospheric Association (NOAA); Central Part weather station in NYC). In the models, as the predictor for the 9am-5pm electricity use, the 9am-5pm average *WBT* ($WBT_{9am\text{-}5pm}$) was then determined by averaging the 9 *WBTs* from 9am to 5pm. Similarly, the predictor for the 12pm-5pm peak demand is the average of the 6 temperatures from 12pm to 5pm ($WBT_{12pm\text{-}5pm}$).

Next, a 7-day moving-average of daily new confirmed Covid-19 cases (henceforth $DCC_{Avg7Day}$) in NYC was used as another independent factor in the regression models. Specifically, for any day for which the electricity consumption was modeled, the factor was the average of the $DCC_{Avg7Day}$ of the previous 7 days, which was obtained from the NYC Department of Health and Mental Hygiene. The previous study in [44] has shown that due to the implementation of the state-level stay-at-home orders after the pandemic, the increased rates of the Covid-19 confirmed cases and time spent at home have a positive correlation of 0.526 (95% confidence interval: 0.293-0.700). Obviously, the implementation of the shelter-in-place restrictions with the large-scale home quarantine, can result in a surge of electricity demand due to more cooking (microwave) and

working (lights, air-conditioners, etc.) at home by residents. Therefore, the daily confirmed cases can be another key factor impacting electricity demand, as it reflects the probability that residents stay at home vs. not (whether out of caution, in response to city-wide guidelines of the "stay-at-home" orders, or both).

As described in Section 3.2.1, most apartments in our dataset consume more electricity in the summer when air conditioners are used, whereas consumption during winter depends only marginally on the weather. Therefore, we developed separate models for times when cooling is not required and times when cooling is required. The threshold temperature (dry-bulb) for requiring cooling versus not in NYC is commonly 18.3°C [20]. Since *WBT* was chosen as the predictor in this work, we converted 18.3°C into its approximate respective *WBT* by using the average of all hourly NOAA-reported *WBT*s measured at times of 18.25-18.34°C in 2019 and 2020. The thus obtained *WBT* threshold (WBT_{thresh}) is 13.8°C.

3.2.3.2 Model structure and rationale

Separate models were devised to forecast the 8-hour-electricity-use on one hand and the 5-hour-peak-demand on the other. Each model was further differentiated into 2 sub-models, one for cooling times and one for non-cooling times, thus yielding a total of 4 separate models.

Inputs, logical flow, and outputs of the 4 models are summarized in **Figure 3.4**. Each of the four models follows two basic steps to predict the electricity usage characteristics during stay-at-home behavior. In step one, the electricity usage data observed in 2019 is used in order to model the two usage characteristics as a function of *WBT* only. This reflects the usage characteristics under a non stay-at-home scenario. In step two, the difference between the observed 2020 usage (observed at a certain *WBT* and $DCC_{Avg7Day}$) and the non-pandemic 2019 usage (modeled for the same *WBT*) is used to devise models to predict the stay-at-home-related increase in electricity

usage. As will be shown below, this increase is a function of $DCC_{Avg7Day}$, and, for outdoor temperatures where cooling is required, also a function of the average *WBT* observed in the daily particular time window for which the electricity usage is predicted.

Note that for the modeling in this case, instead of more complex methods such as neural networks, we opted for traditional multi-factor regression models in order to retain transparency of the mathematical relationships. This approach was chosen in particular to retain robustness of the models when predicting electricity usage for parameter ranges of $DCC_{Avg7Day}$ and *WBT* that had not been observed (see Section 3.2.4). The optimization of coefficients was carried out stepwise: The coefficients for modeling 2019 data and for the single factor transformations were optimized first, and these coefficients were then held constant in the subsequent 2-factor linear regressions. The step-wise optimization of coefficients minimizes the degrees of freedom in each modeling step and thus further reduces any risk of overfitting. Coefficients in all regression models were chosen to minimize the mean squared errors between the observed and the modeled data.

In keeping with this 2-step process, the sections below are therefore organized as follows: Section 3.2.3.3 illustrates the broad relationship between *WBT* and the 8-hour-electricity-use, including the impact of stay-at-home conditions from 2019 to 2020. Section 3.2.3.4 illustrates the same for the 5-hour-peak-demand. Based on these impacts, Sections 3.2.3.5 and 3.2.3.6 then illustrate the details of the modeling process for the 8-hour-electricity-use and 5-hour-peak-demand, by employing single-factor analysis, log and exponential factor transformations, and multi-factor linear regression. Section 3.2.4 provides the equations for combining these models to forecast the 8-hour-electricity-use and 5-hour-peak-demand under a potential future scenario of widespread stay-at-home conditions that also coincide with warm weather. Section 3.2.5 provides the evaluation metric for the models' prediction accuracy.

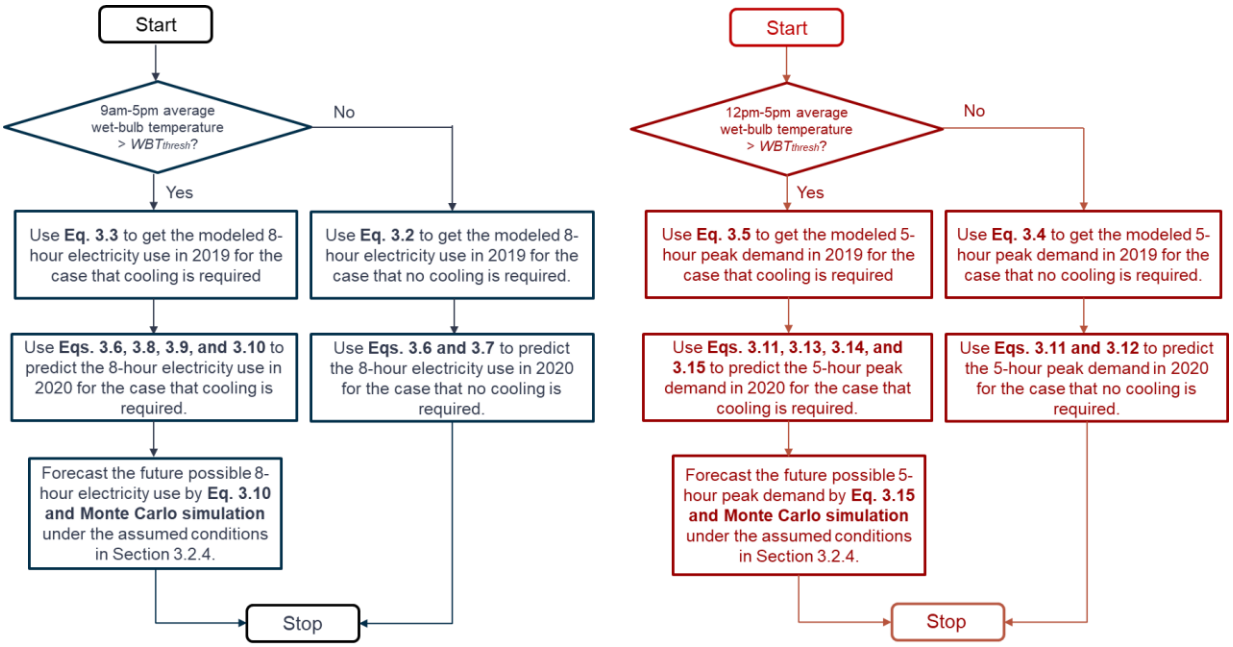

Figure 3.4: Flowcharts of the forecast models for weekday 8-hour-electricity-use (in kWh per average, occupied apartment; left) and 5-hour-peak-demand (in Watt per average, occupied apartment; right).

3.2.3.3 Modeling 2019 usage: 9am-5pm (8-hour) weekday electricity-use

As seen in **Figure 3.5 (a)**, when cooling is not required, *WBT* only marginally impacts the 8-hour-electricity-use, and a straight line with a negative slope thus provides a robust fit:

$$\hat{y}^{(i)}_{use2019} = m_1 WBT_{9am-5pm} + m_2 \tag{3.2}$$

where $\hat{y}^{(i)}_{use2019}$ is the modeled 8-hour-electricity-use in 2019, and $WBT_{9am\text{-}5pm}$ is as above. m_1 and m_2 are the two coefficients of the linear regression. The superscript "(i)" represents the case where cooling is not required (i.e., $WBT_{9am\text{-}5pm}$ smaller than WBT_{thresh} (13.8°C)).

For times when cooling is required, as shown in **Figure 3.5 (b)**, one choice is to model the 8-hour-electricity-use variation with *WBT* to be approximately exponential. We chose an exponential relationship as it provided the best R^2 (compared to using constant and linear regressions, or their combinations) in the temperature range of interest. As introduced in the dataset overview (Section

3.2.1), heating in most apartments does not contribute to the apartments' electricity usage (except for heating blankets or space heaters) but air conditioning does. Therefore, the electricity consumption does not vary significantly with the increase of temperature at a lower temperature range (no cooling required) and implementing an exponential relationship provided a good fit. This fit is defined as follows:

$$\hat{y}_{use2019}^{(ii)} = m_3 e^{m_4 WBT_{9am-5pm}} \tag{3.3}$$

where $\hat{y}_{use2019}^{(ii)}$ is the predicted 8-hour-electricity-use in 2019, and $WBT_{9am-5pm}$ is as above. m_3 and m_4 are the two coefficients of the exponential regression. The superscript "(ii)" represents the case where cooling is required (i.e., $WBT_{9am\text{-}5pm}$ larger than WBT_{thresh} (13.8°C)).

As seen in **Figure 3.5 (c)** and **(d)**, the 8-hour-electricity-use in 2020, both for when cooling is required and not, shows considerable increases vs. 2019, consistent with the diurnal analysis discussed in Section 3.2.2.1. Specifically, we can find from **Figure 3.5 (c)** that during low-temperature periods (below ~5°C), there is no material difference between the 8-hour-electricity-use of the two years. That is consistent with the fact that, in NYC, the COVID-19 pandemic, and thus the associated stay-at-home conditions, only started at the end of winter. In contrast, in warmer weather (above ~5°C), there is a difference in the 8-hour-electricity-use of the two years (indicated by black arrows), and this difference rises exponentially for temperatures above WBT_{thresh} (13.8°C). This indicates that, during the summertime, stay-at-home conditions led to more pronounced increases in the 8-hour-electricity-use in 2020 due to the dominant impact of the higher temperature, even though $DCC_{Avg7Day}$ had decreased at that time and, following gradual relaxing of stay-at-home guidelines, presumably fewer residents were "sheltering-in-place". Again, this is consistent with the result shown in Section 3.2.2.1 (**Figure 3.3**).

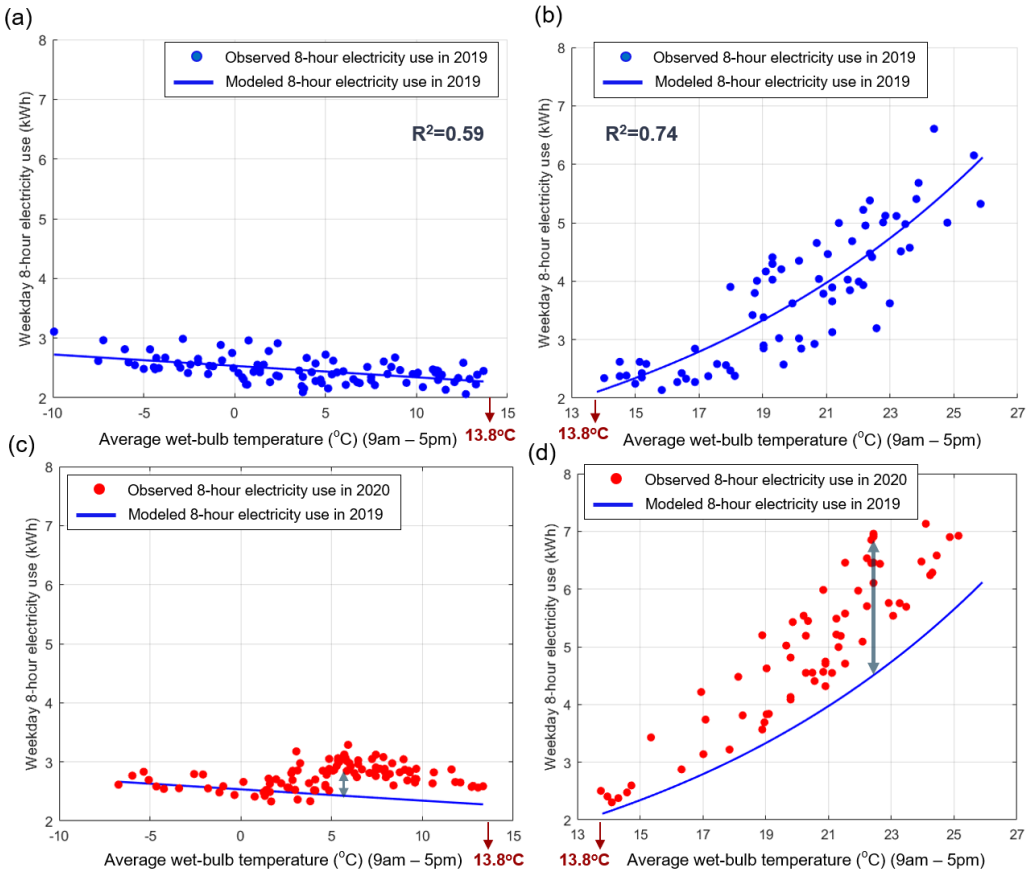

Figure 3.5: Weekday 8-hour apartment electricity usage vs. *WBT*. in 2019 and 2020.

3.2.3.4 Modeling 2019 usage: 12pm-5pm (5-hour) weekday demand peaks

A linear regression and an exponential regression, both based on *WBT*, were set up to model the 5-hour-peak-demand in 2019 (Fig. 8), as follows:

$$\hat{y}_{peak2019}^{(i)} = k_1 WBT_{12pm-5pm} + k_2 \tag{3.4}$$

$$\hat{y}_{peak2019}^{(ii)} = k_3 e^{k_4 WBT_{12pm-5pm}} \tag{3.5}$$

where $WBT_{12pm-5pm}$ is as defined above. $\hat{y}_{peak2019}^{(i)\ or\ (ii)}$ is the modeled 5-hour-peak-demand. k_1, k_2, k_3 and k_4 are the coefficients of the regression. Again, the superscripts "(i)" and "(ii)" denote the two cases of no cooling required and cooling required, respectively. As seen in **Figure 3.6 (d)**, the

5-hour-peak-demand is even more sensitive to temperature fluctuations in warmer weather than the 8-hour-electricity-use (**Figure 3.5**), with implications for grid stability (see Conclusions).

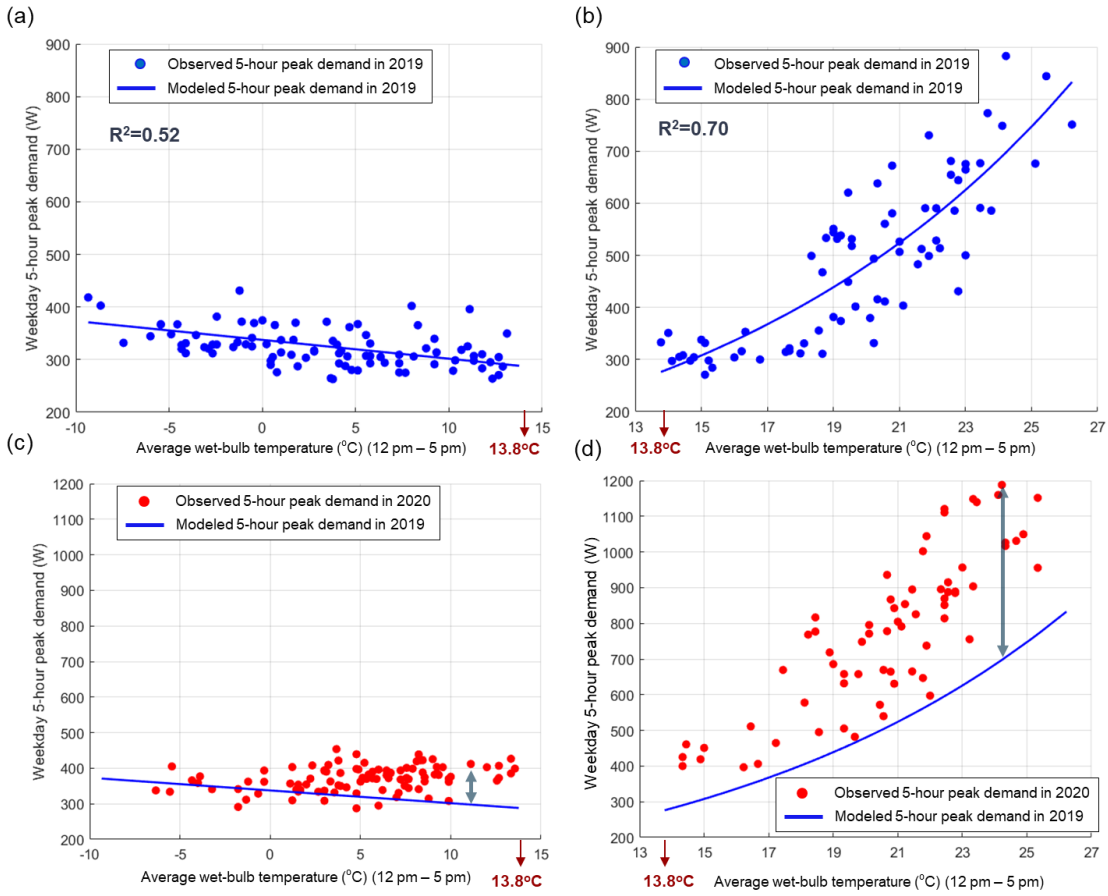

Figure 3.6: Weekday 5-hour apartment peak demand vs. *WBT*. in 2019 and 2020.

3.2.3.5 Predicting increases in usage: 9am-5pm (8-hour) weekday electricity-use

Next, we carried out a series of single-factor analyses to identify a robust model for the increase in weekday 8-hour-electricity-use (9am – 5pm) from 2019 to 2020 as a function of $WBT_{9am\text{-}5pm}$ and $DCC_{Avg7Day}$. As motivated in Section 3.2.2.1, the increase was defined as follows:

$$y_{useinc}^{(i)} = y_{use2020}^{(i)} - \hat{y}_{use2019}^{(i)}$$

$$y_{useinc}^{(ii)} = y_{use2020}^{(ii)} - \hat{y}_{use2019}^{(ii)} \quad (3.6)$$

where $y_{useinc}^{(i)\ or\ (ii)}$ denotes the increases of the 8-hour-electricity-use from 2019 to 2020, each determined as the difference between the observed use in 2020 $y_{use2020}^{(i)\ or\ (ii)}$ and the modeled use in 2019 $\hat{y}_{use2019}^{(i)\ or\ (ii)}$ (modeled for $WBT_{9am\text{-}5pm}$ observed in 2020; see Section 3.2.3.2). The superscripts "(i)" or "(ii)" denote the two cases of no cooling required (N=107 observations) or cooling required (N=67 observations), respectively.

Through the single-factor analysis shown in **Figure 3.7 (a)**, one can find that the increase in 8-hour-electricity-use is logarithmically impacted by $DCC_{Avg7Day}$. As seen in **Figure 3.6 (b)**, the increase resembles a step function as *WBT* rises. However, the step is most likely not principally caused by the *WBT* change but rather by stay-at-home conditions: **Figure 3.5 (c)** shows that the average increase that corresponds to lower *WBTs* (around -6.7°C – 4.5°C) is zero (open blue circles in **Figure 3.7 (b)**). These lower *WBTs* correspond to the period pre-stay-at-home (before the pandemic) from January to February 2020. When the *WBT* reaches about 5°C, the increase in 8-hour-electricity-use is higher (solid blue circles in **Figure 3.7 (b)**), but there are no additional noticeable trends as a function of further increasing *WBT*. Therefore, we set the dependence of the increase in 8-hour-electricity-use on *WBT* to zero. For temperatures when cooling was not required, the final regression model was thus defined as follows:

$$\textbf{Model 1:}\quad \begin{cases} \hat{y}_{useinc}^{(i)} = \beta_{1.1} ln\left(DCC_{Avg7Day} + \beta_{1.2}\right) \\ \hat{y}_{use2020}^{(i)} = \hat{y}_{use2019}^{(i)} + \hat{y}_{useinc}^{(i)} \end{cases} \tag{3.7}$$

where $\hat{y}_{useinc}^{(i)}$ denotes the predicted increase in the 8-hour-electricity-use, and $\hat{y}_{use2020}^{(i)}$ denotes the predicted 8-hour-electricity-use in 2020. $\beta_{1.1}$and $\beta_{1.2}$ are the two coefficients of the regression. The corresponding statistical metrices and modeling performance are shown in **Table 3.2** and **Table 3.6**, respectively.

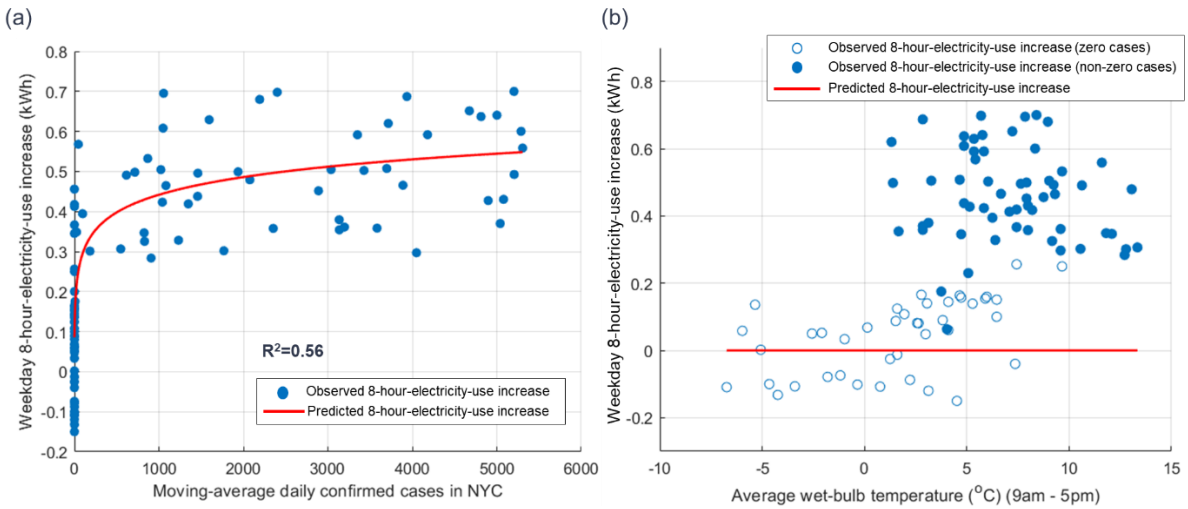

Figure 3.7: (a) Increase in weekday 8-hour-electricity-use (9 am – 5 pm) vs. $DCC_{Avg7Day}$ in NYC. (b) Same vs. $WBT_{9am\text{-}5pm}$.

We first analyzed the relationship between the 8-hour-electricity-use-increase and $DCC_{Avg7Day}$. As shown in **Figure 3.8 (a)**, the data again shows a roughly logarithmic trend. Therefore, to maximize the forecasting accuracy of the subsequent regression model, a logarithmic transformation for the $DCC_{Avg7Day}$ was implemented, as follows:

$$DCC^{log}_{Avg7Day} = max(a_1 ln(DCC_{Avg7Day}) + a_2, 0) \quad (3.8)$$

where $DCC_{Avg7Day}$ as above and $DCC^{log}_{Avg7Day}$ denotes its transformation to be used in the subsequent regression model. a_1 and a_2 are the two coefficients. The maximum operator in Eq. (3.8) sets a zero floor to avoid negative predicted values for electricity usage.

As seen in **Figure 3.8 (a)**, the employed logarithmic transformation does not match data observations ideally, for the following reason: By summer time 2020, $DCC_{Avg7Day}$ in NYC had decreased substantially. This led to the fact that at high-temperatures, when the 8-hour-electricity-use is largely affected by cooling as displayed by the data highlighted by the black dashed circle in **Figure 3.8 (a)**, the observations at high temperatures are not actually at times of high $DCC_{Avg7Day}$. However, when $DCC_{Avg7Day}$ were higher earlier that year, as represented by the data

points highlighted by the black solid circle in **Figure 3.8 (a)**, temperatures were not yet that hot and the corresponding 8-hour-electricity-use thus had not reached its maximum possible values. This re-confirms our observation in Section 3.2.2 that the final regression model for increases in electricity use during widespread stay-at-home conditions must consider both $DCC_{Avg7Day}$ and *WBT*.

As for the relationship between increases in electricity usage and *WBT*, **Figure 3.7 (b)** shows an approximately exponential relationship. We therefore devised an exponential transformation for *WBT*, as follows:

$$WBT_{9am-5pm}^{exp} = b_1 e^{b_2 WBT_{9am-5pm}} \tag{3.9}$$

where $WBT_{9am-5pm}$ in 2020 is as above, and $WBT_{9am-5pm}^{exp}$ is its exponential transformation to be used in the subsequent linear regression. b_1 and b_2 are the two coefficients. The two transformed variables $DCC_{Avg7Day}^{log}$ and $WBT_{9am-5pm}^{exp}$ were then used as the two independent variables in a two-factor linear regression model for predicting the 8-hour-use-increase when cooling is required, as follows:

$$\textbf{Model 2:} \quad \begin{cases} \hat{y}_{useinc}^{(ii)} = \beta_{2.1} + \beta_{2.2} DCC_{Avg7Day}^{log} + \beta_{2.3} WBT_{9am-5pm}^{exp} \\ \hat{y}_{use2020}^{(ii)} = \hat{y}_{use2019}^{(ii)} + \hat{y}_{useinc}^{(ii)} \end{cases} \tag{3.10}$$

where $\hat{y}_{useinc}^{(ii)}$ denotes the predicted increase in 8-hour-electricity-use, and $\hat{y}_{use2020}^{(ii)}$ denotes the predicted 8-hour-electricity-use in 2020. $DCC_{Avg7Day}^{log}$ and $WBT_{9am-5pm}^{exp}$ are as defined above, and $\beta_{2.1}$, $\beta_{2.2}$, and $\beta_{2.3}$ are the three coefficients of the 2-factor linear regression model, whose statistical metrics and modeling performance are shown in **Table 3.3** and **Table 3.6**, respectively.

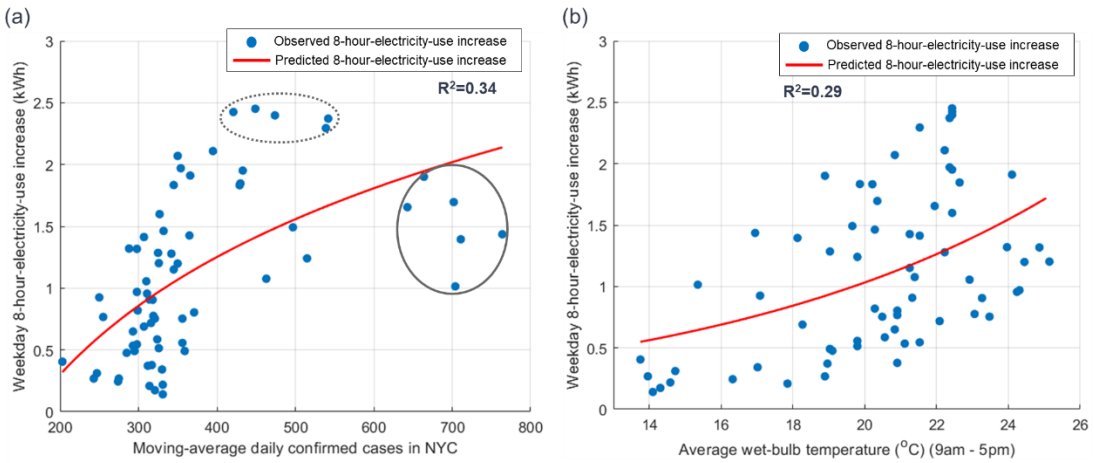

Figure 3.8: (a) Increase in weekday 8-hour-electricity-use (9am – 5pm) vs. *DCC*$_{Avg7Day}$. (b) Same vs. *WBT*$_{9am}$-5pm. All data points are for times when cooling is required during Jan – Aug.

3.2.3.6 Predicting increases in usage: 12pm-5pm (5-hour) weekday peak demands

Next, we used similar methods to analyze and forecast the weekday 5-hour-peak-demand (12pm – 5pm) as a function of the two factors (*WBT*$_{12pm\text{-}5pm}$ and *DCC*$_{Avg7Day}$). The increase was defined as follows:

$$y_{peakinc}^{(i)} = y_{peak2020}^{(i)} - \hat{y}_{peak2019}^{(i)}$$

$$y_{peakinc}^{(i)} = y_{peak2020}^{(i)} - \hat{y}_{peak2019}^{(i)} \quad (3.11)$$

where $y_{peakinc}^{(i)\ or\ (ii)}$ denotes the increases of the 5-hour-peak-demand from 2019 to 2020, each determined as the difference between the observed peak demand in 2020 $y_{peak2020}^{(i)\ or\ (ii)}$ and the modeled peak demand in 2019 $\hat{y}_{peak2019}^{(i)or\ (ii)}$ (modeled for the respective *WBT*$_{12pm\text{-}5pm}$ observed in 2020; see Section 3.2.3.2). Again, the superscripts "(i)" or "(ii)" represent the two cases of no cooling required (N=105 observations) and cooling required (N=69 observations), respectively.

For *DCC*$_{Avg7Day}$, **Figure 3.9 (a)** reveals an approximately logarithmic trend, similar to the one for increases in 8-hour-electricity-use in **Figure 3.7 (a)**. The relationship with *WBT*$_{12pm\text{-}5pm}$ shown

in **Figure 3.8 (b)** is similar to a step function, as above. Therefore, we chose again to set the dependence of the increases in 5-hour-peak-demand on $WBT_{12pm\text{-}5pm}$ to zero. The final model is as follows:

$$\textbf{Model 3:} \quad \begin{cases} \hat{y}_{peakinc}^{(i)} = \beta_{3.1}\, ln\left(DCC_{Avg7Day} + \beta_{3.2}\right) \\ \hat{y}_{peak2020}^{(i)} = y_{peak2019}^{(i)} + \hat{y}_{peakinc}^{(i)} \end{cases} \tag{3.12}$$

where $\hat{y}_{peakinc}^{(i)}$ denotes the predicted increase in the 5-hour-peak-demand, and $\hat{y}_{peak2020}^{(i)}$ denotes the predicted 5-hour-peak-demand in 2020. $\beta_{3.1}$ and $\beta_{3.2}$ are the two coefficients of the logarithmic regression model, and $DCC_{Avg7Day}$ is as above. The corresponding statistical metrics and modeling performance are shown in **Table 3.4** and **Table 3.6**, respectively.

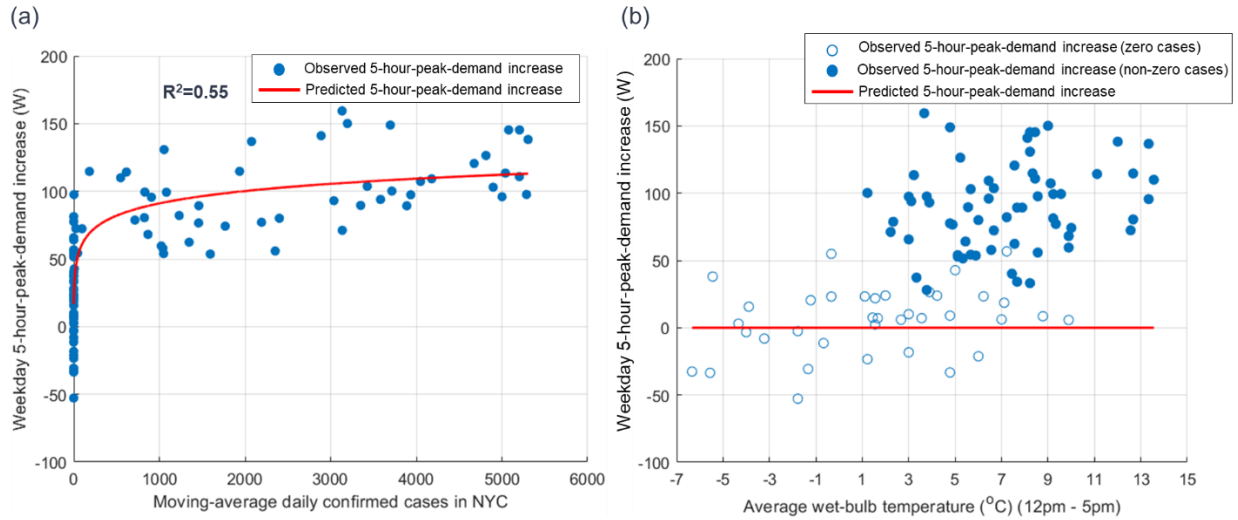

Figure 3.9: (a) Increase in weekday 5-hour-peak-demand (12 pm – 5 pm) vs. $CDD_{Avg7Day}$. (b) Same vs. $WBT_{12pm\text{-}5pm}$. Data points are for times when cooling is not required.

The relationships for increases in 5-hour-peak-demand in **Figure 3.10** are similar to what we described for the increases in 8-hour-electricity-use: (i) When cooling is required, only considering $DCC_{Avg7Day}$ is not sufficient to predict the increases. Instead, the impact of $WBT_{12pm\text{-}5pm}$ must be considered as well; (ii) a logarithmic and exponential transformation can be used to maximize the

forecasting accuracy of the subsequent linear regression model. The factor transformations were as follows:

$$DCC_{Avg7Day}^{log} = max(c_1 ln(DCC_{Avg7Day}) + c_2,\ 0) \tag{3.13}$$

$$WBT_{12pm-5pm}^{exp} = d_1 e^{d_2 WBT_{12pm-5pm}} \tag{3.14}$$

where $DCC_{Avg7Day}$, $DCC_{Avg7Day}^{log}$, $WBT_{12pm-5pm}$ and $WBT_{12pm-5pm}^{exp}$ are as defined above. c_1, c_2, d_1, and d_2 are the coefficients of the log and exponential transformations. The maximum operator in Eq. (3.13) sets a zero floor for the transformation so that the subsequent regression model does not yield negative predicted values.

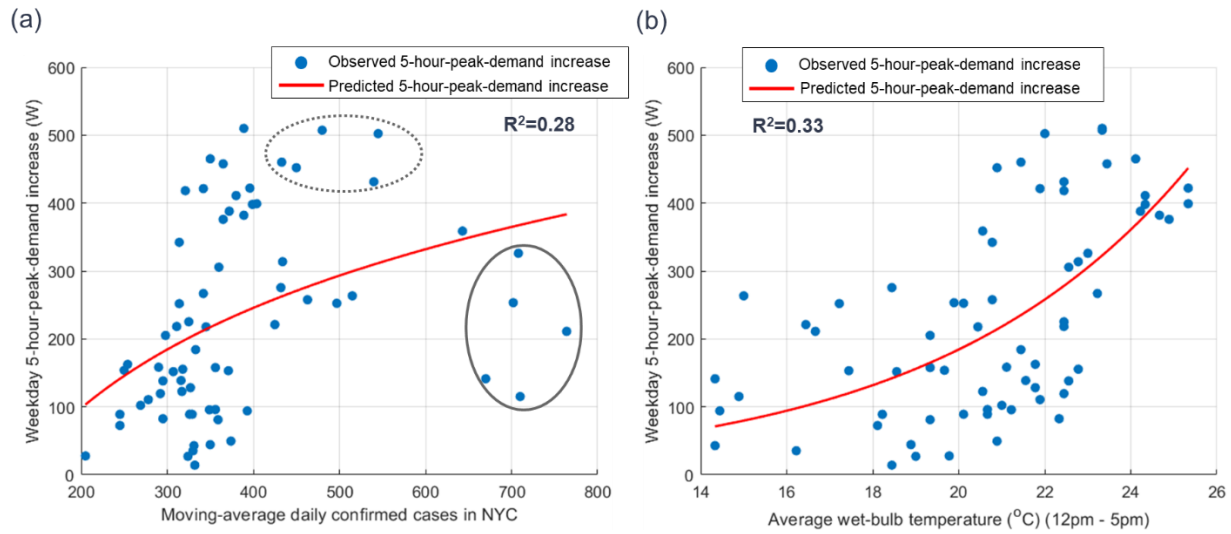

Figure 3.10: (a) Increase in weekday 5-hour-peak-demand (12 pm – 5 pm) vs. $CDD_{Avg7Day}$. (b) Same vs. $WBT_{12pm-5pm}$. Data points are for times when cooling is required.

Next, the two transformed variables $DCC_{Avg7Day}^{log}$ and $WBT_{12pm-5pm}^{exp}$ were used as independent variables in a two-factor linear regression model to forecast the increase in 5-hour-peak-demand, as follows:

$$\textbf{Model 4:}\ \begin{cases} \hat{y}_{peakinc}^{(ii)} = \beta_{4.1} + \beta_{4.2} DCC_{Avg7Day}^{log} + \beta_{4.3} WBT_{12pm-5pm}^{exp} \\ \hat{y}_{peak2020}^{(ii)} = y_{peak2019}^{(ii)} + \hat{y}_{peakinc}^{(ii)} \end{cases} \quad (3.15)$$

where, $\hat{y}_{peakinc}^{(ii)}$ denotes the predicted increase in the 5-hour-peak-demand, and $\hat{y}_{peak2020}^{(ii)}$ denotes the predicted 5-hour-peak-demand in 2020. $\beta_{4.1}$, $\beta_{4.2}$, and $\beta_{4.3}$ are the three coefficients of the 2-factor linear regression model, whose statistical metrics and modeling performance are shown in **Table 3.5** and **Table 3.6**, respectively.

3.2.4 Monte Carlo simulation for possible extreme future scenario

Our ultimate objective is to predict the possible values of 8-hour-electricity-use and 5-hour-peak-demand in the future, if widespread stay-at-home behavior (due to a worsening pandemic or other reasons) and warm weather were to coincide in NYC. We chose a simulation for this rather than the directly observed data itself, for the following reason: In 2020, NYC did not experience a scenario when high $DCC_{Avg7Day}$ coincided with high *WBT*. Rather, in April, when the daily case numbers were at their highest, the *WBT* in NYC was still below the value of WBT_{thresh}, and air conditioning did not yet take place at any material rate. When *WBT* rose in June and July, the impacts of the pandemic in NYC had eased, and people were no longer required to comply with the stay-at-home guidelines (known as phase one and phase two reopening). There is therefore no directly observable electricity usage data for the putative "worst case" scenario of high $DCC_{Avg7Day}$ (and thus a high portion of residents working/studying from home) combined with high temperatures.

For such a prediction, we extracted those observed values of the two predictors ($DCC_{Avg7Day}$ and *WBT*) that met the assumed conditions separately and recombined them to create a new dataset via simulation, as follows: We selected only the subset of observed $DCC_{Avg7Day}$ that were greater

than half of its Jan.-Dec. 2020 maximum (i.e., greater than 2,651) and only the *WBT* that were greater than WBT_{thresh}. Then, in a Monte Carlo simulation [39], we randomly sampled 1,000 times from the two extracted subsets to generate a new set of predictive factors consisting of 1,000 pairs of data (each pair with one value for $DCC_{Avg7Day}$ and one value for *WBT*). The simulated factors were then used in Eq. (3.10) and Eq. (3.15) to predict the corresponding 1,000 predictions for 8-hour-electricity-usage (kWh) and the 1,000 predictions for 5-hour-peak-demand (Watt).

3.2.5 Evaluation metric for prediction accuracy

In order to assess the prediction accuracy of the four models (Model 1 – 4), we compared the predicted values of the 8-hour-electricity-use and 5-hour-peak-demand to the respective values observed in 2020 using the common R^2 metric (coefficient of determination):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.16}$$

where y_i is the observed 8-hour-electricity-use or 5-hour-peak-demand in 2020, and $\widehat{y_i}$ is the corresponding predicted value, and the corresponding evaluation results of the four models are shown in **Table 3.6**. It should be noted that the R^2 results in **Figure 3.5-Figure 3.10** are the intermediary evaluation results of the single-factor regressions needed in the stepwise modeling process, which do not reflect the accuracies of the four models as evaluated by Eq. (3.16).

3.3 Results

3.3.1 Model calibration and prediction accuracy

As outlined in the section of Data and Methods, we used a set of four models to predict the two electricity usage characteristics we focused on in this case: (i) The cumulative 9am-5pm

electricity usage (in kWh) for the average apartment on weekdays (henceforth "8-hour-electricity-use"); and (ii) the highest hourly peak demand (in Watt) for the average apartment in the hours of 12pm-5pm (henceforth "5-hour-peak-demand"). The two independent variables used in each prediction are (i) the 7-day rolling average of daily confirmed Covid-19 case numbers in NYC prior to the day of observed electricity use ($DCC_{Avg7Day}$); and (ii) the outdoor *WBT* averaged over the respective time window on the day of observed electricity use, $WBT_{9am\text{-}5pm}$ or $WBT_{12pm\text{-}5pm}$.

Specifically, Model 2 and Model 1 predict the 8-hour-electricity-use, separately for the two cases when cooling is required or not, respectively. Model 4 and Model 3 predict the 5-hour-peak-demand for the same two cases. The regression coefficients and their 95% confidence intervals for all models are provided in **Table 3.2-Table 3.5**. **Figure 3.11** displays the predicted and observed 8-hour-electricity-use and 5-hour-peak-demand in 2020. The prediction accuracies, assessed as R^2 separately for each of the four models, are shown in **Table 3.6**.

Overall, the models enable robust predictions of the two electricity usage characteristics in 2020, with R^2 from 0.56 to 0.84 (**Table 3.6**). However, differences in accuracy between the 4 models exist. The prediction accuracy is higher at higher temperatures of $WBT > 13.8°C$ (R^2 of 0.84 and 0.80 for Models 2 and 4) than the accuracy at smaller temperatures when no air conditioning is required (R^2 of 0.57 and 0.56 for Models 1 and 3). The more accurate regime is key to determining whether there are potential challenges and risks for electricity grids (see Conclusions). Another, but less pronounced difference is that, within the high temperature regime, the model to predict the 8-hour-electricity-use (R^2=0.84 for Model 2) is moderately more accurate than the model for the 5-hour-peak-demand (R^2=0.80 for Model 4). This is also reflected in the narrower 95% confidence intervals of the respective model coefficients. It is possibly due to more volatile/idiosyncratic cooling loads during the summertime. For the conclusions of this paper

(Section 3.4), they are reached by the results of the Model 2 and Model 4 (the high-temperature case that cooling is required), which have promising accuracy with the R^2 of 0.84 and 0.80, respectively.

Table 3.2: Coefficients for Model 1 (prediction of the 8-hour-electricity-use when cooling is not required). 95% confidence intervals of the coefficients are reported in parentheses.

	m_1	m_2	$\beta_{1.1}$	$\beta_{1.2}$
Results	-0.019 (-0.022, -0.016)	2.535 (2.513, 2.557)	0.0641 (0.059, 0.069)	3.828 (1.409, 6.248)
p values	4.15e-08	2.36e-10	2.21e-4	7.56e-8

Table 3.3: Coefficients for Model 2 (prediction of the 8-hour-electricity-use when cooling is required.). 95% confidence intervals of the coefficients are reported in parentheses.

	m_3	m_4	a_1	a_2	b_1	b_2	$\beta_{2.1}$	$\beta_{2.2}$	$\beta_{2.3}$
Results	0.625 (0.447, 0.803)	0.088 (0.075, 0.101)	1.377 (0.911, 1.843)	-6.998 (-9.744, -4.255)	0.137 (0.023, 0.297)	0.101 (0.047, 0.154)	-1.151 (-1.408, -0.883)	0.978 (0.839, 1.117)	1.058 (0.875, 1.241)
p values	3.12e-5	4.11e-13	7.26e-5	1.12e-4	8.55e-4	3.24e-8	3.99e-8	8.25e-9	6.14e-9

Table 3.4: **Coefficients for Model 3 (prediction of the 5-hour-peak-demand when cooling is not required.). 95% confidence intervals of the coefficients are reported in parentheses.**

	k_1	k_2	$\beta_{3.1}$	$\beta_{3.2}$
Results	-3.578 (-4.168, -2.988)	337.6 (333.5, 341.6)	13.17 (12.11, 14.24)	3.556 (1.282, 5.830)
p values	4.88e-6	2.71e-7	3.12e-3	4.11e-7

Table 3.5: **Coefficients for Model 4 (prediction of the 5-hour-peak-demand when cooling is required.). 95% confidence intervals of the coefficients are reported in parentheses.**

	k_3	k_4	c_1	c_2	d_1	d_2	$\beta_{4.1}$	$\beta_{4.2}$	$\beta_{4.3}$
Results	81.38 (57.86, 104.9)	0.088 (0.075, 0.102)	212.9 (93.94, 331.8)	-1030 (-1743, -325.6)	6.426 (9.711, 3.451)	0.1678 (0.1123, 0.2233)	-248.9 (-305.1, -192.7)	1.0963 (0.8973, 1.2953)	0.9969 (1.1196, 0.8742)
p values	6.72e-5	1.19e-12	9.61e-4	4.47e-5	3.75e-3	7.22e-4	3.58e-5	6.64e-7	1.77e-11

Table 3.6: **Model accuracy determined from the observed and predicted 8-hour-electricity-use and 5-hour-peak-demand in 2020. N denotes the number of data points for each R^2 statistic.**

	Model 1: 8-hour-electricity-use without cooling	Model 2: 8-hour-electricity-use with cooling	Model 3: 5-hour-peak-demand without cooling	Model 4: 5-hour-peak-demand with cooling
R^2	0.57	0.84	0.56	0.80
N	107	67	105	69

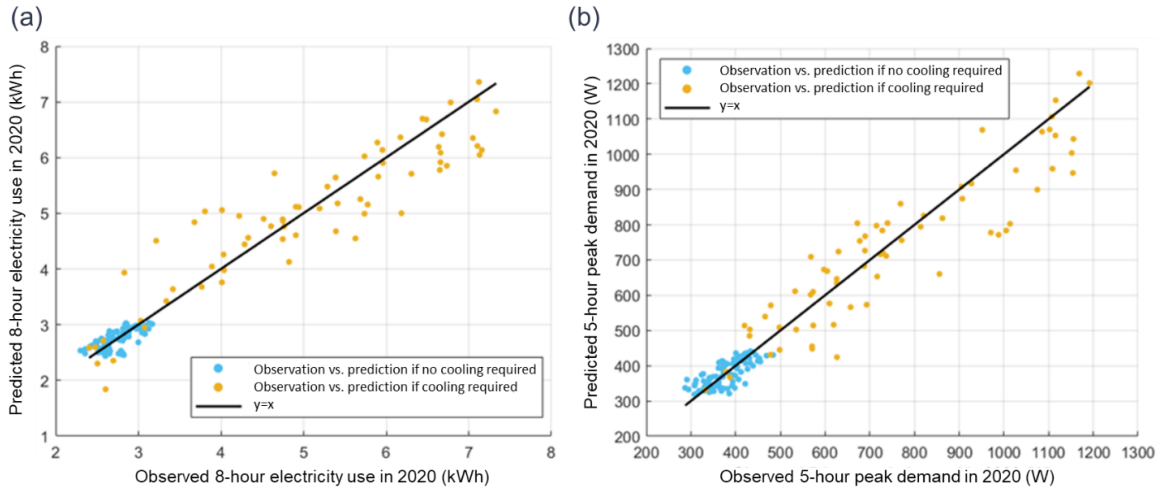

Figure 3.11: Model performance. (a) Observed vs. predicted 8-hour-electricity-usage in 2020. (b) Same for 5-hour-peak-demand.

3.3.2 Forecasting the two usage characteristics in a hypothetical future scenario

Finally, the models were applied to predict the possible 8-hour-electricity-use and 5-hour-peak-demand in a scenario in which both warm weather and widespread stay-at-home behavior – due to (for example) a renewed, severe level of the pandemic – might coincide in NYC or similar metropolitan areas in the future. As shown in the preliminary analyses in Section 3.2.2, there are no *observed* data points for the combined condition, where *WBT* is larger than WBT_{thresh} (13.8°C) and $DCC_{Avg7Day}$ is larger than 2,651 (half of the maximum $DCC_{Avg7Day}$ observed in Jan.-Aug. 2020). For such a scenario, the Monte Carlo simulation (Section 3.2.4) was employed to generate new data satisfying the respective conditions, and there are two main advantages for using such an

approach. First, when it comes to the two predictors for the hypothetical scenario, namely the *WBT* larger than WBT_{thresh} and the $DCC_{Avg7Day}$ larger than 2651, both of them do not follow a normal distribution by referring the results of the Kolmogorov-Smirnov (K-S) test [53] (both generate the p-value smaller than 0.05, which rejects the null hypothesis that the statistical distribution is same as Gaussian.). Therefore, a Monte Carlo simulation is likely to generate more realistic data for the two predictors, instead of simply using the averages of the two for prediction, so as to obtain a more reliable range of the forecasting results instead of a single predicted value. In addition, as the Model 2 and 4, developed through the logarithm and exponential transformations for the *WBT* and $DCC_{Avg7Day}$ are nonlinear, a more realistic dataset produced by the Monte Carlo simulation can take into account the nonlinear relationship between the two predictors and the increases, which thereby enables more accurate forecasting for the hypothetical scenario.

The corresponding predicted future-possible 8-hour-electricity-use and 5-hour-peak-demands are shown in **Table 3.7**. The Monte Carlo simulations show that, for the average, occupied apartment, the 8-hour-electricity-use and 5-hour-peak-demand are likely to be 7.63—8.21 kWh and 1211—1369 Watts, respectively. Note that this is an estimate spanning a range of conditions where *WBT* is larger than 13.8^{o}C and $DCC_{Avg7Day}$ is larger than 2,651. As seen in **Figure 3.6**, the highest observed 8-hour-electricity-use and 5-hour-peak-demand in 2019 were 6.61 kWh and 894 Watts respectively. Compared to these observed values, we therefore predict that the 8-hour-electricity-use could be 15%—24% higher than the one under normal circumstances (pre-stay-at-home period), and the 5-hour-peak-demand could be 35% – 53% higher.

Table 3.7: Predicted results of the 8-hour-electricity use and 5-hour-peak-demand, generated using Monte Carlo simulations with Model 2 and Model 4 respectively, with values for ± 1 standard deviation in parentheses.

	Predicted results in 2020	Maximum observed in 2019	Estimated percentage increase ranges
8-hour electricity use (kWh)	7.92 (8.21, 7.63)	6.61	15%—24%
5-hour peak demand (W)	1289 (1369, 1211)	894	35%—53%

Large *WBT* values could lead to a potential rapid rise of the 5-hour-peak-demand, and we thus further explored the observed and predicted 5-hour-peak-demand under the various $DCC_{Avg7Day}$ scenarios and $WBT_{12pm\text{-}5pm}$ observed in Jul.-Aug., the warmest summer months (**Figure 3.12**). One observes that when $WBT_{12pm\text{-}5pm}$ is constant, the 5-hour-peak-demand increases logarithmically with the increase in the number of $DCC_{Avg7Day}$, as stated in the established Model 4 (Section 3.2.3.6). Observe that the maximum 5-hour-peak-demand in 2019 was 894 Watts at $WBT_{12pm\text{-}5pm}$ of 24.2^{o}C and 0 cases, and the maximum observed value in 2020 was 1,188 Watts at $WBT_{12pm\text{-}5pm}$ of 24.4 ^{o}C and $DCC_{Avg7Day}$ of 396. The green band illustrated in **Figure 3.12** corresponds to the projected peak for the highest-case load band of between 2,651 and 5,301 of $DCC_{Avg7Day,}$ if these cases were to occur during the hotter temperatures shown here that require cooling. The projected 5-hour-peak-demand could certainly exceed the maximum observed one in 2019 (894 Watts), and at hotter temperatures could be twice as high as the corresponding 2019 peak, potentially leading to new risks for electrical grids in the future (see Conclusions). The peak demand for any hour in 2019 was observed to be up to 983 Watts, which, without stay-at-home orders, commonly occurs only in the late evenings over the summer. The projected green band also exceeds this peak by a wide margin.

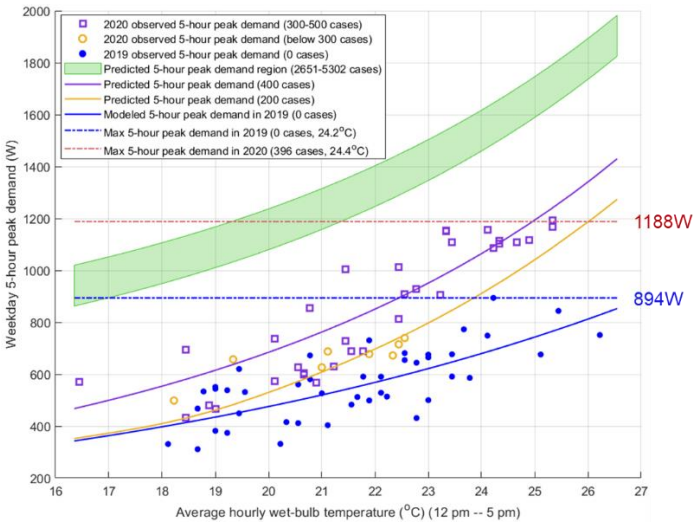

Figure 3.12: Observed and predicted weekday 5-hour-peak-deamd (12pm-5pm, per apartment) in Jul.-Aug. 2019 and 2020 under the scenarios of various $DCC_{Avg7Day}$.

3.4 Conclusions

Comparing 2020 with 2019 residential electricity consumption data, a case study was conducted to investigate and forecast Covid-19-related increases in residential electricity usage of occupied apartments in NYC, based on a sample of 390 apartments. The apartments are, in size and vintage, representative of NYC residential building stock, and their electricity consumption is consistent with other multi-family settings in the same climate region. We focused on two characteristics of residential electricity usage, (i) the electricity consumption (kWh) of an average apartment on weekdays in the 8 hours from 9am to 5pm (in order to gauge shifts in energy use and commensurate financial burdens from commercial buildings and schools to the residential sector); and (ii) the hourly peak demand (Watt) of an average apartment in the 5 hours between 12pm and 5pm (in order to gauge possible stress on the electricity grid when this peak either coincides with system-wide loads or becomes larger than what feeders and distribution lines in residential areas were designed to handle).

We identified two factors and built a series of regression models which can predict the above two characteristics with an R^2 of 0.56-0.57 for days when no cooling is required and 0.80-0.84 for warmer days. The two factors are the severity of the pandemic (measured as a 7-day rolling average of daily confirmed Covid-19 cases in NYC) and the outdoor *WBT* (measured as the average *WBT* during the respective 8-hour or 5-hour window). The models indicate that increases in residential electricity usage between 2019 and 2020 were the higher, the more severe the pandemic (which we interpret as a proxy for the portion of residents working and studying from home). And for times when cooling was required, these increases were further modulated by the outdoor temperature. Therefore, in NYC in 2020, usage increases versus 2019 continued to grow more pronounced during the summer months even while lockdown measures were being partially lifted.

In a Monte Carlo simulation, we then used the models to forecast the two usage characteristics for conditions which, fortunately, did not actually occur in 2020, but which could occur in the future in NYC, in similar regions, or indeed in future pandemics or natural catastrophes with comparable stay-at-home guidelines. These conditions were the combination of high outdoor wet bulb temperatures (such that cooling in the apartments is required) coupled with medium to high pandemic severity (and with it a high presumed portion of residents working or studying from home).

We found that under such assumed future conditions, the weekday 8-hour-electricity-use (9am-5pm) could be 15%—24% higher than the one under normal circumstances (i.e., no stay-at-home behavior), implying a corresponding substantial increase in electricity costs for residents.

We further found that the weekday 5-hour-peak-demand (12pm-5pm) could be 35%—53% higher than otherwise. This suggests possible grid stress especially if substantial increases in residential demand coincide with recovery in commercial demand. At high daily case numbers

(100% of Jan-Aug. 2020 maximum) and $WBT_{12pm\text{-}5pm}$ above 25°C, the 5-hour hourly peak demand would be nearly twice that of the maximum 5-hour-peak-demand in 2019 (894 Watts). It would also be much higher than the largest-ever observed peak demand in 2019 (983 Watts). In predominantly residential network areas and feeders with no commensurate load reduction in commercial buildings to offset this increase, such high peaks – nearly twice as high as the prior year peak – could lead to loads that exceed the designed feeder capacity, possibly leading to failure risks of the local substation and distribution infrastructure.

This chapter can provide a meaningful reference point for building managers and utilities to improve the balance of supply and demand in future grids, for example through battery storage in residential buildings, distributed storage, market-mechanisms to encourage the integration of grid-efficient interactive buildings into smart grids, including via storage in electric vehicles, and Transactive Energy Networks [16]. In such contexts, the models introduced in this chapter could be integrated with emerging smart-grid management techniques, in order to improve the residential electricity forecasting accuracy under stay-at-home guidelines due to a pandemic or other natural catastrophes – or to account for the potential of a "new-normal" lifestyle, even in the absence of a catastrophe.

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE), under the Building Technologies Office's BENEFIT program, Award Number DE-EE-0007864.

Chapter 4. A New Generalized Autoencoder for Structural Damage Assessment

The main part of this chapter is presented in the paper co-authored with Dr. Marcello Morgantini and Prof. Raimondo Betti, and published in the Journal of Mechanical Systems and Signal Processing [54].

4.1 Introduction

In recent years, advances in sensors and computer technologies have supported various promising developments in structural-health-monitoring (SHM) techniques. Data obtained from sensors installed on a structure can help engineers continuously assess the structural integrity, reduce the operational costs, and optimize the available resources [55]. In dealing with buildings and bridges, the most common measurements available for such analyses is represented by the time histories of the structural response, i.e., acceleration and/or displacement, recorded at different locations on the structure in service conditions or during particular single events (e.g., an earthquake or hurricane). Because of the nature of the data used, these SHM methodologies fall into the category of vibration-based SHM approaches.

Among all the possible features used in damage assessment strategies that rely on the vibration-based SHM approaches, modal characteristics (e.g., modal frequencies, mode shapes, modal damping ratios, etc.), which are functions of the physical parameters of the structure (mass, damping, and stiffness), have been proven to be very effective and practical features for structural damage assessment [56]. For example, Shih et al. [57] developed a multi-criteria-based non-destructive procedure to detect damage in a slab-on-girder bridge, by accounting for changes in natural frequencies, modal flexibility, and modal strain energy. They concluded that the modal

flexibility and the modal strain energy can reliably identify the scenarios of single and multiple damages in the bridge's girders and deck. Going beyond the changes in modal characteristics, an alternative strategy for damage assessment is the one based on model updating [58], where the recorded responses of a real-life system are used to update iteratively the physical parameters of a mathematical model (usually a finite element model) until the model can accurately reproduce the recorded responses: If there are substantial variations in the physical parameters during the monitoring period, e.g. a noticeable drop in stiffness, it might indicate that the system has suffered some structural damages.

In recent years, thanks to the advances in modern computer performances, features extracted directly from the structural responses through simple digital signal processing tools have become very appealing since their extraction is very fast and does not require large computational resources and great expertise by users. Among these data-based features, the ones defined in the cepstrum domain, extensively used in the fields of speech and speaker recognition, have been proven to be quite effective in SHM applications [59]. The cepstrum of a signal, originally defined as the "power spectrum of the logarithm of the power spectrum", was first introduced by Bogert et al. [60], when they developed a method to detect echoes from time-series signals. Recently, cepstrum-based features have been employed in structural damage assessment by Zhang et al. [61]. They used Mel-Frequency Cepstral Coefficients (MFCCs) to characterize the bridge deck acoustic response to ultrasonic pulses to study the delamination of the concrete deck. In 2014, Balsamo et al. [59] used the MFCCs obtained from the vibration response of buildings and bridges as Damage Sensitive Features (DSFs), with a novelty detection strategy integrated with statistical-pattern-recognition analysis. In 2021, Morgantini et al. [55] presented a theoretical investigation that shows analytically how the power cepstral coefficients of the structural acceleration responses are

linked to the modal characteristics of the structure and how they can be successfully used in a damage assessment strategy. One of the advantages of using these cepstral coefficients is represented by the rapidity with which they can be extracted from the original time signal, compared to other features (e.g., natural frequencies) that require complex and time-consuming operations. This characteristic of the cepstral coefficients, together with their intrinsic connection with the structure's modal characteristics, has served as the springboard for the development of the new generalized auto-encoder, presented in this chapter, for the rapid assessment of structural damage.

With the recent explosion of Machine Learning (ML) applications in every sector of our life, ML and deep-learning techniques are finding fertile grounds in many civil engineering applications and recently have shown great potential in structural damage assessment [62], thanks also to an increasingly large amount of measurement data from real buildings and bridges. Among these techniques used in SHM problems, the convolutional neural networks (CNNs) [63], implemented on a supervised-learning framework, have shown promising results when used in damage assessment in concrete and steel structures [64, 65]. However, even though supervised strategies can provide fairly accurate damage assessment results, they need proper network training that commonly requires large datasets representative of both the undamaged structure and the structure in different damage conditions, a requirement that cannot be obviously satisfied when dealing with real-life structures [66].

Consequently, over recent years, many studies have focused on approaches of unsupervised learning for the damage assessment in buildings and bridges. Pathirage et al. [67] developed an unsupervised-learning framework for structural damage assessment, which consists of a deep autoencoder for structural characteristics dimension reduction, and a simple autoencoder for a

regression task of predicting structural stiffness reduction. Through numerical and experimental investigations on steel frame structures, they concluded that the proposed framework enables improved accuracy and efficiency in structural damage assessment compared to the traditional neural network approaches. Ma et al. [68] used a Variational Auto-Encoder (VAE) to learn a compressed hidden representation of the structural acceleration responses to be used in damage detection. Through the numerical studies of a beam-like bridge, they concluded that the proposed method can accurately identify different types of damages that were simulated by setting various crack depths on the structure. Along the same line of research, Wang et al. [66] proposed an unsupervised approach, based on a deep auto-encoder and on a one-class support vector machine, to assess structural damage, using the recorded acceleration responses of the intact structures as training data: the proposed method enabled high assessment accuracy (91% or higher).

Although the use of appropriate features can effectively improve the performance of ML models in the task of assessing damage, there are still some unavoidable bottlenecks in modeling with many deep-learning algorithms, e.g., the overly complicated network structure, leading to slow training speed and overfitting issues without reasonable model generalization [69]. In this chapter, a New Generalized Auto-Encoder (NGAE), integrated with a statistical-pattern-recognition strategy and the power cepstral coefficients of the recorded structural acceleration responses, is proposed for structural damage assessment. This NGAE is capable of capturing the component of the power cepstral coefficients that is linked to the overall structural properties and, at the same time, of shrinking the data variance caused by different external excitations, sensor and actuator locations, and measurement noise. The cepstral coefficients, by virtue of a compact representation of the structural properties, can greatly simplify the structure of the network, and therefore, significantly accelerate both the training and the inference speeds, compared to an auto-

encoder that uses the recorded acceleration responses or the traditional modal features as the network's inputs and outputs [66, 67]. Based on the well-trained NGAE, two evaluation metrics for assessing damage are computed and integrated with a statistical-pattern-recognition approach for further damage detection and quantification. The effectiveness of the proposed method was validated by both simulated and real-life examples, comparing the NGAE's results with those obtained using a Traditional Auto-Encoder (TAE) [70] and those obtained through the Principal Component Analysis (PCA) [71].

4.2 Methodology

4.2.1 Analytical expression of the cepstral coefficients of structural acceleration

The power cepstral coefficients, extracted from the acceleration response of a structure, provide an alternative and compact representation of modal properties of the structural system (e.g., natural frequencies, damping ratios and mode shapes) and have shown great potential in structural damage assessment.

Let us consider the equations of motion of an N_d degree-of-freedom (DOF) model of a linear time-invariant system:

$$\boldsymbol{M}\ddot{\boldsymbol{y}}(t) + \boldsymbol{C}\dot{\boldsymbol{y}}(t) + \boldsymbol{K}\boldsymbol{y}(t) = \boldsymbol{u}(t) \tag{4.1}$$

where $\boldsymbol{M} \in \mathbb{R}^{N_d \times N_d}$, $\boldsymbol{C} \in \mathbb{R}^{N_d \times N_d}$ and $\boldsymbol{K} \in \mathbb{R}^{N_d \times N_d}$ are the mass, damping and stiffness matrices, respectively, each of dimension $N_d \times N_d$. The vector $\boldsymbol{y}(t) \in \mathbb{R}^{N_d}$ is the vector of nodal displacement, $\dot{\boldsymbol{y}}(t) \in \mathbb{R}^{N_d}$ the nodal velocity vector, and $\ddot{\boldsymbol{y}}(t) \in \mathbb{R}^{N_d}$ the nodal acceleration vector. $\boldsymbol{u}(t) \in \mathbb{R}^{N_d}$ is the input vector, containing the values of the nodal external excitations at time t.

For the general case where all the N_d DOFs are excited by N_d different input excitation, considering that the measured structural response comes in as a discrete time signal, the z-transform of the acceleration time history at the d^{th} DOF ($d = 1, \dots, N_d$), $A_d(z)$, can be then expressed as:

$$A_d(z) = \sum_{j=1}^{N_d} H_a(z)_{d,j} U_j(z) \tag{4.2}$$

where $H_a(z)_{d,j}$ represents the $(d,j)^{\text{th}}$ term of the inertance matrix and $U_j(z)$ represents the z-transform of the input excitation $\boldsymbol{u}_j(t)$ applied at the j^{th} DOF ($j = 1, \dots, N_d$). By expanding the right-hand side of Eq. (2), $A_d(z)$ can be rewritten in the form of products as:

$$A_d(z) = \frac{(1 - z^{-1}) \prod_{l=1}^{M} (1 - Z_l^{(d)} z^{-1})}{\prod_{l=1}^{N} (1 - e^{\lambda_l \Delta t} z^{-1})(1 - e^{\lambda_l^* \Delta t} z^{-1})} \tag{4.3}$$

where λ_l and λ_l^* are the complex conjugate eigenvalues associated with the l^{th} vibrational mode of the system ($l = 1, \dots, N_d$), and Δt represents the sampling time interval at which the structural acceleration has been recorded. The symbols $Z_l^{(d)}$ for $l = 1, \dots, M$ are the M roots of the following equation:

$$\sum_{j=1}^{N_d} U_j(z) \sum_{l=1}^{N_d} \phi_{d,l} \phi_{j,l} (1 - P_{a,l} z^{-1}) \prod_{\substack{k=1 \\ k \neq l}}^{N_d} \left(1 - e^{\lambda_k \Delta t} z^{-1}\right)\left(1 - e^{\lambda_k^* \Delta t} z^{-1}\right) = 0 \tag{4.4}$$

where $\phi_{d,l}\phi_{j,l}$ are the components of the l^{th} mode shape at the corresponding d^{th} and j^{th} locations. $P_{a,l}$ indicates a function of a modal characteristics and of the type of measurements considered. As shown in [1], Eq. (4.4) accounts for the locations of the forcing functions and of the sensors, for the type of measurement as well as for the type and magnitude of the input forces. By taking the Inverse Discrete Fourier Transform (IDFT) of the logarithm of the squared

magnitude of $A_d(z)$, the cepstral coefficients extracted from the time history of the acceleration recorded at the d^{th} DOF can be expressed as follows:

$$c_d[q] = \frac{1}{q}\left[\sum_{l=1}^{N_d} 2e^{-\xi_l \omega_l \Delta t q} \cos\left(\omega_{damp,l}\Delta t q\right) - 1 - \sum_{l=1}^{M} Z_l^{(d)q}\right] \quad \text{for } q > 0 \tag{4.5}$$

where $c_d[q]$ represents q^{th} cepstral coefficient for the acceleration at the d^{th} DOF ($d = 1, \dots, N_d$) with q indicating the "quefrency" index. Only the cepstral coefficients for $q > 0$ are considered, as the $c_d[q]$ at $q = 0$ depends only on the sampled input while the $c_d[q]$ for $q < 0$ are simply equal to zero [1]. The parameters ξ_l and ω_l are the damping ratio and natural frequency associated with the l^{th} vibrational mode of the system, respectively, and $\omega_{damp,l} = \omega_l\sqrt{1-\xi_l^2}$ is the corresponding damped natural frequency. For a detailed derivation of Eq. (4.5), the reader is referred to [1].

In this chapter, it is important to note that the expression of the cepstral coefficients of the structural acceleration recorded at the d^{th} location (Eq. (4.5)) can be re-written as:

$$c_d[q] = \theta[q] + \gamma_d[q] \tag{4.6}$$

where $\theta[q]$ and $\gamma_d[q]$ are given by:

$$\begin{cases} \theta[q] = \dfrac{1}{q}\displaystyle\sum_{l=1}^{N_d} 2e^{-\xi_l \omega_l \Delta t q} \cos\left(\omega_{damp,l}\Delta t q\right) - 1 \\ \gamma_d[q] = -\dfrac{1}{q}\displaystyle\sum_{l=1}^{M} Z_l^{(d)q} \end{cases} \tag{4.7}$$

Eq. (4.6) and (4.7) offer some important insights into the nature of the cepstral coefficients. It appears that the cepstral coefficients $c_d[q]$ ($q = 1, 2, \dots, Q$) can be thought as composed by two terms, namely $\theta[q]$ and $\gamma_d[q]$. The term $\theta[q]$ only depends on the structural properties (natural frequencies and damping ratios) of the overall structural system and thus it is independent of the

location where the structural acceleration has been recorded, i.e., the same $\theta[q]$ is present in all q^{th} cepstral coefficients extracted from the acceleration responses recorded at different locations on the structure. On the contrary, the component $\gamma_d[q]$, which is completely related to the roots $Z_l^{(d)}$, depends on the recording location as well as on the locations where the forcing functions are applied (through the components of the mode shapes at those location), on the characteristics of the external excitations, and on the overall structural properties. Hence, it is reasonable to expect that, for the cepstral coefficients extracted from the recorded acceleration responses of a system, the larger contribution to the variance in the estimation of the cepstral coefficients comes from the term $\gamma_d[q]$, while the contribution from $\theta[q]$ should remain basically constant, except for some inevitable measurement noise. Therefore, it would be helpful to develop an effective strategy for damage assessment that reduces the variance caused by the excitation-related term $\gamma_d[q]$ and by measurement noise and enhances the weight of the term $\theta[q]$.

4.2.2 Autoencoders and the proposed framework

4.2.2.1 The traditional autoencoder

As shown in Section 4.2.1, the cepstral coefficients, extracted from the structural acceleration, might present a large variance as a result of the different external excitations and this might hide potential changes in the $\theta[q]$ counterpart induced by damage. In order to strictly assess a structure's state without considerable interference by the excitations and measurement noise, building an autoencoder-based model to characterize the underlying structural properties (embedded in the cepstral coefficients) can be an effective solution.

The autoencoder is one type of unsupervised neural networks, which commonly sets its output values equal to its inputs through backpropagation of numerical optimization [70]. A wide variety

of autoencoder-based models have been used for representation learning and feature dimension reduction, handling large amounts of unlabeled recorded data [72]. The simplest structure of a traditional autoencoder (TAE) consists of an input layer, a hidden layer, and an output layer, where the input and its corresponding output should be identical to each other, as shown in **Figure 4.1**. Alternatively, a TAE can be considered as a two-part system, namely an encoder and a decoder, where the encoder maps an input vector into a compressed hidden representation, while the decoder maps the hidden representation to a reconstruction of the original input.

Let's first investigate the case where a single-hidden-layer TAE is set up using the power cepstral coefficients of the structural acceleration as its inputs and outputs. Let's assume that a structure in its undamaged state is monitored at N_d locations and that a training dataset $\{\boldsymbol{x}_{1,d}, \dots, \boldsymbol{x}_{N_{tr},d}\}_{d=1}^{N_d}$ has been created. Such a dataset accounts for $N_{tr} \times N_d$ vectors (commonly called instances) $\boldsymbol{x}_{i,d} \in \mathbb{R}^m$ ($i = 1, \dots, N_{tr}$ with $N_{tr} > m$, and $d = 1, \dots, N_d$), where each $\boldsymbol{x}_{i,d}$ contains the Q cepstral coefficients extracted from the i^{th} record of the acceleration response at the d^{th} recording location. Thus, the vector $\boldsymbol{x}_{i,d}$ can be expressed as:

$$\boldsymbol{x}_{i,d} = \{c_{i,d}[1],\ c_{i,d}[2],\ \dots c_{i,d}[q], \dots, c_{i,d}[Q]\}^T \tag{4.8}$$

where q represents the q^{th} element in the vector $\boldsymbol{x}_{i,d}$. Using the Mean Squared Error (MSE) between the input and the reconstructed output of the TAE as the loss function of the TAE, we can state the optimization problem of training a TAE built to model the structural response at the d^{th} recording location as follows:

$$\begin{aligned}&[\boldsymbol{W}_{1,d},\ \boldsymbol{W}_{2,d}, \boldsymbol{b}_{1,d}, \boldsymbol{b}_{2,d}]\\ &= \arg\ \min \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \frac{1}{Q} \left\| \boldsymbol{x}_{i,d} - g\left(\boldsymbol{W}_{2,d}\left(f\left(\boldsymbol{W}_{1,d}\boldsymbol{x}_{i,d} + \boldsymbol{b}_{1,d}\right) + \boldsymbol{b}_{2,d}\right)\right)\right\|_F^2\end{aligned} \tag{4.9}$$

where $\boldsymbol{W}_{1,d} \in \mathbb{R}^{p\times Q}$ and $\boldsymbol{W}_{2,d} \in \mathbb{R}^{Q\times p}$ are the weights of the encoder and decoder, respectively, and $\boldsymbol{b}_{1,d} \in \mathbb{R}^{p}$ and $\boldsymbol{b}_{2,d} \in \mathbb{R}^{Q}$ are the corresponding biases, with Q and p representing the dimensions of the input/output (Q) and hidden spaces (p), i.e., the input/output and hidden layer sizes, respectively, and generally $p < Q$. In Eq. (4.9), the operator $\|\cdot\|_F^2$ represents the Frobenius norm. The encoded hidden representation $\boldsymbol{h}_{i,d} \in \mathbb{R}^{p}$ by the encoder, and the reconstructed output $\hat{\boldsymbol{x}}_{i,d} \in \mathbb{R}^{Q}$ by the decoder can be expressed by:

$$\begin{aligned} \boldsymbol{h}_{i,d} &= f\left(\boldsymbol{W}_{1,d}\boldsymbol{x}_{i,d} + \boldsymbol{b}_{1,d}\right) \\ \hat{\boldsymbol{x}}_{i,d} &= g\left(\boldsymbol{W}_{2,d}\boldsymbol{h}_{i,d} + \boldsymbol{b}_{2,d}\right) \end{aligned} \tag{4.10}$$

where $f(\cdot)$ and $g(\cdot)$ represent the element-wise activation functions for the encoder and decoder, respectively (usually the sigmoid function or hyperbolic tangent function [20]). Generally, the nonlinearity of the activation functions $f(\cdot)$ and $g(\cdot)$ enables high-level computational abilities, but can lead to a higher level of difficulty in solving the optimization problem. In this work, the Adam optimization algorithm [73] has been employed in the training of the TAE. By solving the optimization problem of Eq. (4.9), the single-hidden-layer TAE, when incorporating a linear or a sigmoid activation function at the hidden layer, is able to learn the underlying information from the input in a similar fashion as when using the PCA [74]. Specifically, the reconstructed output of such a TAE is strongly related to the PCA reconstruction based on the first p principal components of the covariance matrix $\boldsymbol{C}_d = \boldsymbol{X}_d\boldsymbol{X}_d^T \in \mathbb{R}^{Q\times Q}$, where $\boldsymbol{X}_d = [\boldsymbol{x}_{1,d}, \dots, \boldsymbol{x}_{N_{tr},d}]$.

However, this structure of the TAE does not perform sufficiently well in detecting damage when using the cepstral coefficients as inputs to the network (as shown in the Numerical Results). In fact, the nature of this TAE would aim to retain as much information as possible from the entire cepstral coefficients (rather than only from the most relevant information about the structural properties contained in $\theta[q]$) and would try to fit the excitation-related variance embedded in

$\gamma_d[q]$, as well as the one caused by the measurement noise. Therefore, it is necessary to generalize the structure of the autoencoder to reach a better characterization of the overall structural properties of the system indicative of the presence of damage.

4.2.2.2 The proposed new generalized autoencoder

The generalized autoencoder (GAE) was first proposed by Wang et al. [75], with the aim of better learning the underlying structure of the original data and of obtaining a well-generalized compressed representation at the hidden space. As originally presented in [75], the GAE is set to establish a mapping that forces each input instance to reconstruct a set of instances based on a relational loss function, rather than reconstruct itself. However, the GAE as originally defined is not applicable in our case, as the defined relational loss function still cannot weaken the contribution to the variance by the term $\gamma_d[q]$ and by the measurement noise. In addition, it is noteworthy that the GAE does not support a strict ground truth (desired output) for each of the input instances, which is different from the TAE whose desired outputs are identical to the corresponding inputs. Therefore, it would be difficult to reasonably assess the signal reconstruction error of the GAE by specific evaluation metrics.

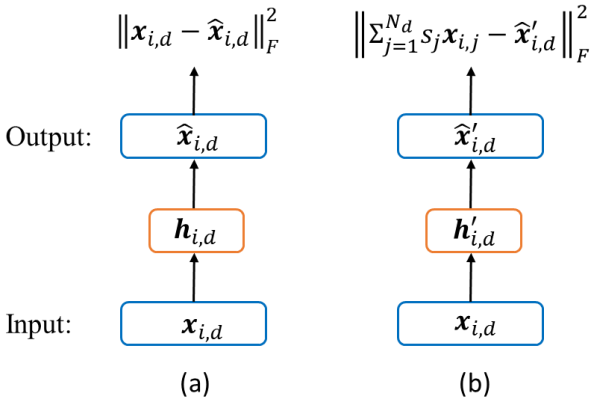

Figure 4.1: The traditional autoencoder (a) and the proposed new generalized autoencoder (b).

By drawing on the idea of the relational loss function of the GAE and on the advantage of the MSE-based signal reconstruction error, a New Generalized Auto-Encoder (NGAE) is proposed for improving modeling performance. **Figure 4.1** offers a comparison between the fundamental mechanisms of the TAE and the NGAE architectures. Starting from the initial dataset, the optimization problem of training a NGAE built to model the structural response at the d^{th} recording location can be expressed as:

$$\begin{aligned}
&\left[\boldsymbol{W}'_{1,d},\ \boldsymbol{W}'_{2,d}, \boldsymbol{b}'_{1,d}, \boldsymbol{b}'_{2,d}\right] \\
&= \arg\min \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \frac{1}{Q} \left\| \boldsymbol{x}'_i - g\left(\boldsymbol{W}'_{2,d}\left(f\left(\boldsymbol{W}'_{1,d}\boldsymbol{x}_{i,d} + \boldsymbol{b}'_{1,d}\right) + \boldsymbol{b}'_{2,d}\right)\right)\right\|_F^2 \\
&= \arg\min \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \frac{1}{Q} \left\| \boldsymbol{x}'_i - g\left(\boldsymbol{W}'_{2,d}\boldsymbol{h}'_{i,d} + \boldsymbol{b}'_{2,d}\right)\right\|_F^2 \\
&= \arg\min \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \frac{1}{Q} \left\| \boldsymbol{x}'_i - \hat{\boldsymbol{x}}'_{i,d}\right\|_F^2
\end{aligned} \tag{4.11}$$

where $\boldsymbol{W}'_{1,d} \in \mathbb{R}^{p\times Q}$, $\boldsymbol{W}'_{2,d} \in \mathbb{R}^{Q\times p}$, $\boldsymbol{b}'_{1,d} \in \mathbb{R}^{p}$ and $\boldsymbol{b}'_{2,d} \in \mathbb{R}^{Q}$ represent the weights and biases of the NGAE. The vector $\boldsymbol{h}'_{i,d} = f\left(\boldsymbol{W}'_{1,d}\boldsymbol{x}_{i,d} + \boldsymbol{b}'_{1,d}\right) \in \mathbb{R}^{p}$ represents the i^{th} encoded hidden representation by the NGAE's encoder, i.e., the i^{th} output of the hidden layer, with p being the hidden space dimension, while the vector $\hat{\boldsymbol{x}}'_{i,d} = g\left(\boldsymbol{W}'_{2,d}\boldsymbol{h}'_{i,d} + \boldsymbol{b}'_{2,d}\right) \in \mathbb{R}^{Q}$ represents the i^{th} reconstructed output by the NGAE's decoder, where $f(\cdot)$ and $g(\cdot)$ represent the element-wise activation functions as introduced in Section 4.2.2.1. The vector $\boldsymbol{x}'_i \in \mathbb{R}^{Q}$, termed as the "new-ground-truth" vector of the NGAE, is set to be the i^{th} desired output of the NGAE. It is defined as a weighted summation of the cepstral coefficient vectors at every recording location of the system, expressed as:

$$\boldsymbol{x}_i' = \sum_{j=1}^{N_d} s_j \boldsymbol{x}_{i,j} \tag{4.12}$$

where s_j is a specific weighting term corresponding to the j^{th} location ($j = 1, \dots, N_d$). Such a coefficient is defined by looking at the average of the variance of the cepstral coefficients in the vectors $\boldsymbol{x}_{i,j}$ ($i = 1, \dots, N_{tr}$) and can be expressed as:

$$s_j = \frac{C}{\frac{1}{Q}\sum_{q=1}^{m} \frac{1}{N_{tr}-1} \sum_{i=1}^{N_{tr}} \left(x_{i,j}[q] - \bar{x}_j[q]\right)^2} \tag{4.13}$$

subject to the condition:

$$\sum_{j=1}^{N_d} s_j = 1 \tag{4.14}$$

where $x_{i,j}[q]$ represents the q^{th} element in the vector $\boldsymbol{x}_{i,j}$ ($q = 1, .., Q$), i.e., the cepstral coefficient $c_{i,j}[q]$, while $\bar{x}_j[q]$ represents the mean of the elements $x_{i,j}[q]$ for $i = 1, \dots, N_{tr}$ with respect to the j^{th} location. The constant C is determined by the equality constraint of Eq. (4.14). Using Eq. (4.6), the new-ground-truth vector $\boldsymbol{x}_i'$ can be expressed as:

$$\boldsymbol{x}_i' = \left\{\theta_i[1] + \textstyle\sum_{j=1}^{N_d} s_j \gamma_{i,j}[1], \dots, \theta_i[q] + \sum_{j=1}^{N_d} s_j \gamma_{i,j}[q], \dots, \theta_i[Q] + \sum_{j=1}^{N_d} s_j \gamma_{i,j}[Q]\right\}^T \tag{4.15}$$

while its corresponding input $\boldsymbol{x}_{i,d}$ is:

$$\boldsymbol{x}_{i,d} = \left\{\theta_i[1] + \gamma_{i,d}[1], \dots, \theta_i[q] + \gamma_{i,d}[q], \dots, \theta_i[Q] + \gamma_{i,d}[Q]\right\}^T \tag{4.16}$$

One can observe that the new-ground-truth vector $\boldsymbol{x}_i'$ contains Q newly-defined cepstral coefficients, each of which maintains intact the contribution $\theta_i[q]$ related to the overall structural properties, while the contribution from the excitation and sensor locations appears as a weighted average of all the corresponding terms at the various locations. Two important points are noteworthy here: First, the newly defined vector $\boldsymbol{x}_i'$ is independent of the locations where the

structural acceleration has been recorded. This implies that all the N_d NGAEs, used to model the overall system, will have the same new-ground-truth vectors $\boldsymbol{x}_i'$ $(i = 1, \dots, N_{tr})$ as the desired outputs for their training. Therefore, if N_d NGAEs are set up for modeling the structural response at the N_d locations in the system, they will have the same decoder mapping after a well training process. The other important point is about the components of $\boldsymbol{x}_i'$ representing the weighted summation of the excitation-related term $\gamma_{i,j}[q]$ for $j = 1, \dots, N_d$. As the weight s_j is set inversely proportional to the variance of all the cepstral coefficients extracted at the j^{th} location, the summation $\sum_{j=1}^{N_d} s_j \gamma_{i,j}\,[q]$ can help shrink the data variance associated with the excitation and measurement location terms, indirectly enhancing the contribution of the term $\theta_i[q]$: This will facilitate the assessment of damage. Consequently, such a mapping from an input $\boldsymbol{x}_{i,d}$ to the corresponding desired output $\boldsymbol{x}_i'$ can be interpreted as building a "stronger" connection between the input and output through their common part $\theta_i[q]$ based on the learned hidden representation $\boldsymbol{h}_{i,d}'$. To visualize this effect, **Figure 4.2** shows the output vector $\boldsymbol{x}_{i,d}$ $(d = 1)$ (introduced by Eq. (4.16)) and the vector $\boldsymbol{x}_i'$ (introduced by Eq. (4.15)) obtained from an 8 DOF shear-type system that will be discussed in Section 4.3.1, subjected to an excitation applied either at the 1st or the 8th DOF. It directly shows that the cepstral coefficients in the vector $\boldsymbol{x}_{i,d}$ (the TAE's desired output, shown in **Figure 4.2 (a)**) produces different trends and distributions under the two different excitation locations (due to different contributions of the term $\gamma_{i,d}[q]$), while the trends and distributions of the modified cepstral coefficients in the vector $\boldsymbol{x}_i'$ (the NGAE's desired output, shown in **Figure 4.2 (b)**) are very close, as the variance in each term of $\boldsymbol{x}_i'$ has been largely reduced.

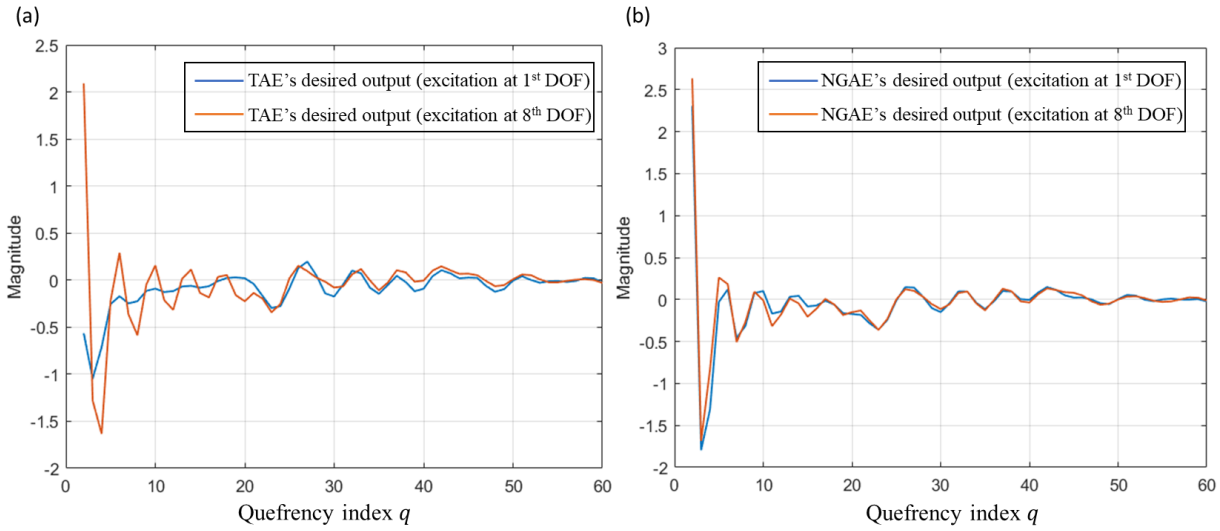

Figure 4.2: Visualizations of the sampled instances $x_{i,d}$ (d = 1) and x'_i obtained from an 8DOF shear-type system, with a zero-mean Gaussian white noise excitation applied at either the 1st or 8th DOF.

4.2.2.3 NGAE with linear activation function

In order to provide a more explicit intuition of the theoretical mechanism of the NGAE, let us now consider the case where both activation functions $f(\cdot)$ and $g(\cdot)$ in Eq. (4.11) are linear, i.e., $f\left(\boldsymbol{W}'_{1,d}\boldsymbol{x}_{i,d} + \boldsymbol{b}'_{1,d}\right) = \boldsymbol{W}'_{1,d}\boldsymbol{x}_{i,d} + \boldsymbol{b}'_{1,d}$ and $g\left(\boldsymbol{W}'_{2,d}\boldsymbol{h}'_{i,d} + \boldsymbol{b}'_{2,d}\right) = \boldsymbol{W}'_{2,d}\boldsymbol{h}'_{i,d} + \boldsymbol{b}'_{2,d}$. In this case, it is possible to derive an analytical solution for the optimization problem associated with the NGAE, defined in Eq. (4.11), so to better understand the mechanism behind the proposed NGAE.

By substituting the above linear activation functions, removing the scaling terms, and applying the properties of the Frobenius norm, Eq. (4.11) can be expressed in a matrix form as:

$$[\boldsymbol{W}'_{1,d},\ \boldsymbol{W}'_{2,d}, \boldsymbol{b}'_{1,d}, \boldsymbol{b}'_{2,d}] = \arg\min \left\| \boldsymbol{X}' - \left(\boldsymbol{W}'_{2,d}\left(\boldsymbol{W}'_{1,d}\boldsymbol{X}_d + \boldsymbol{b}'_{1,d}\mathbf{1}^T_{N_{tr}}\right) + \boldsymbol{b}'_{2,d}\mathbf{1}^T_{N_{tr}}\right)\right\|_F^2 \qquad (4.17)$$

where the columns of the matrices $\boldsymbol{X}' \in \mathbb{R}^{Q\times N_{tr}}$ and $\boldsymbol{X}_d \in \mathbb{R}^{Q\times N_{tr}}$ are the desired output vectors $\boldsymbol{x}'_i \in \mathbb{R}^Q$ and input vectors $\boldsymbol{x}_{i,d} \in \mathbb{R}^Q$ $(i = 1, \dots, N_{tr})$, respectively, and $\mathbf{1}_{N_{tr}} \in \mathbb{R}^{N_{tr}}$ is a vector of ones. Let us define $\boldsymbol{H}'_d = \boldsymbol{W}'_{1,d}\boldsymbol{X}_d + \boldsymbol{b}'_{1,d}\mathbf{1}^T_{N_{tr}}$ $(\boldsymbol{H}'_d \in \mathbb{R}^{p\times N_{tr}})$ as the hidden output matrix of the

NGAE, whose columns are the hidden outputs $\boldsymbol{h}'_{i,d}$ $(i = 1, \dots, N_{tr})$. Such a new matrix will allow us to express Eq. (4.17) as:

$$\begin{aligned}\left[\boldsymbol{W}'_{1,d},\ \boldsymbol{W}'_{2,d}, \boldsymbol{b}'_{1,d}, \boldsymbol{b}'_{2,d}\right] &= \arg\ \min \left\|\boldsymbol{X}' - \left(\boldsymbol{W}'_{2,d}\boldsymbol{H}'_d + \boldsymbol{b}'_{2,d}\mathbf{1}^T_{N_{tr}}\right)\right\|_F^2 \\ &= \arg\ \min \left\|\boldsymbol{X}' - \widehat{\boldsymbol{X}}'_d\right\|_F^2\end{aligned} \tag{4.18}$$

where $\widehat{\boldsymbol{X}}'_d = \boldsymbol{W}'_{2,d}\boldsymbol{H}'_d + \boldsymbol{b}'_{2,d}\mathbf{1}^T_{N_{tr}}$ $(\widehat{\boldsymbol{X}}'_d \in \mathbb{R}^{Q\times N_{tr}})$ indicates the reconstructed output matrix of the NGAE at the d^{th} location. To determine the "optimal" weights $\boldsymbol{W}'_{1,d}$ and $\boldsymbol{W}'_{2,d}$, and biases $\boldsymbol{b}'_{1,d}$ and $\boldsymbol{b}'_{2,d}$, for the NGAE of the d^{th} location, let us first set the partial derivative of Eq. (18) with respect to $\boldsymbol{b}'_{2,d}$ equal to zero: This will provide an expression for the "optimal" $\boldsymbol{b}'_{2,d}$ as:

$$\boldsymbol{b}'_{2,d} = \frac{1}{N_{tr}}\left(\boldsymbol{X}' - \boldsymbol{W}'_{2,d}\boldsymbol{H}'_d\right)\mathbf{1}_{N_{tr}} \tag{4.19}$$

Substituting the solution of $\boldsymbol{b}'_{2,d}$ given by Eq. (4.19) and $\boldsymbol{H}'_d = \boldsymbol{W}'_{1,d}\boldsymbol{X}_d + \boldsymbol{b}'_{1,d}\mathbf{1}^T_{N_{tr}}$ into Eq. (18), the optimization problem can be re-written in a more concise form as:

$$\left[\boldsymbol{W}'_{1,d},\ \boldsymbol{W}'_{2,d}, \boldsymbol{b}'_{1,d}\right] = \arg\min \left\|\widetilde{\boldsymbol{X}}' - \boldsymbol{W}'_{2,d}\widetilde{\boldsymbol{H}}'_d\right\|_F^2 \tag{4.20}$$

where $\widetilde{\boldsymbol{X}}' = \boldsymbol{X}'(\boldsymbol{I} - \mathbf{1}_{N_{tr}}\mathbf{1}^T_{N_{tr}}/N_{tr})$ and $\widetilde{\boldsymbol{H}}'_d = \boldsymbol{H}'_d(\boldsymbol{I} - \mathbf{1}_{N_{tr}}\mathbf{1}^T_{N_{tr}}/N_{tr})$, which are basically the matrices $\boldsymbol{X}'$ and $\boldsymbol{H}'_d$ subtracted by their element-wise averages. It is important to note that the hidden space dimension p determines the rank of $\boldsymbol{W}'_{2,d} \in \mathbb{R}^{Q\times p}$ because $p < Q$. From Eq. (20), the matrix multiplication $\boldsymbol{W}'_{2,d}\widetilde{\boldsymbol{H}}'_d$ that minimizes the loss function can be obtained from the truncated Singular Value Decomposition (SVD) [35] of the matrix $\widetilde{\boldsymbol{X}}'$:

$$\widetilde{\boldsymbol{X}}' \approx \boldsymbol{U}'_p\boldsymbol{\Sigma}'_p{\boldsymbol{V}'}^T_p = \boldsymbol{W}'_{2,d}\widetilde{\boldsymbol{H}}'_d \tag{4.21}$$

where the columns of $\boldsymbol{U}'_p \in R^{Q\times p}$ and $\boldsymbol{V}'_p \in R^{N_{tr}\times p}$ are formed by the first p normalized eigenvectors of the $\widetilde{\boldsymbol{X}}'{\widetilde{\boldsymbol{X}}'}^T$ and ${\widetilde{\boldsymbol{X}}'}^T\widetilde{\boldsymbol{X}}'$, respectively, associated with the first p eigenvalues $\lambda'_1 \geq \lambda'_2 \geq \cdots \geq \lambda'_p \geq 0$. The matrix $\boldsymbol{\Sigma}'_p = diag[\sigma'_1, \sigma'_2, \dots, \sigma'_p]$ is a diagonal matrix that contains the first

p singular values of the matrix $\widetilde{\boldsymbol{X}}'$ with $\sigma_s' = \sqrt{\lambda_s'}$ $(s = 1, \dots, p)$. Therefore, by setting the hidden layer size p equal to the rank $r_{\boldsymbol{X}'}$ of the desired output matrix $\boldsymbol{X}'$, the multiplication $\boldsymbol{W}'_{2,d}\widetilde{\boldsymbol{H}}'_d$ can theoretically achieve the best-rank-$r_{\boldsymbol{X}'}$ approximation of $\widetilde{\boldsymbol{X}}'$ and, consequently, the best-rank-$r_{\boldsymbol{X}'}$ approximation of $\boldsymbol{X}'$, given the relationship between $\widetilde{\boldsymbol{X}}'$ and $\boldsymbol{X}'$, i.e., $\widetilde{\boldsymbol{X}}' = \boldsymbol{X}'(\boldsymbol{I} - \mathbf{1}_{N_{tr}}\mathbf{1}_{N_{tr}}^T/N_{tr})$. Such a property is also applicable when building a single-hidden-layer TAE [32]. The "optimal" solutions of $\boldsymbol{W}'_{2,d}$ and $\widetilde{\boldsymbol{H}}'_d$ in Eq. (4.21) can then be expressed as:

$$\boldsymbol{W}'_{2,d} = \boldsymbol{U}'_p{\boldsymbol{T}'_p}^{-1}, \quad \widetilde{\boldsymbol{H}}'_d = \boldsymbol{T}'_p\boldsymbol{\Sigma}'_p{\boldsymbol{V}'_p}^T \tag{4.22}$$

where $\boldsymbol{T}'_p \in R^{p\times p}$ is a non-singular matrix that generally cannot be eliminated through the backpropagation process during the training operation, leading to a nonorthogonal learned hidden space for the NGAE. Finally, taking advantages of Eq. (4.19) and (4.21) and considering $\boldsymbol{H}_d = \widetilde{\boldsymbol{H}}_d(\boldsymbol{I} - \mathbf{1}_{N_{tr}}\mathbf{1}_{N_{tr}}^T/N_{tr})^{-1}$, the optimal reconstructed output $\widehat{\boldsymbol{X}}'_d$, expressed as $\widehat{\boldsymbol{X}}'_d = \boldsymbol{W}'_{2,d}\boldsymbol{H}'_d + \boldsymbol{b}'_{2,d}\mathbf{1}_{N_{tr}}^T$, can be obtained as:

$$\widehat{\boldsymbol{X}}'_d = \boldsymbol{U}'_p\boldsymbol{\Sigma}'_p{\boldsymbol{V}'}_p^T + \overline{\boldsymbol{X}}' \tag{4.23}$$

where $\overline{\boldsymbol{X}}'$ is equal to $\boldsymbol{X}'\mathbf{1}_{N_{tr}}\mathbf{1}_{N_{tr}}^T/N_{tr}$, representing the element-wise averages of $\boldsymbol{X}'$.

In conclusion, the optimal $\widehat{\boldsymbol{X}}'_d$ allows us to derive a NGAE that can capture as much structural-property information as possible from the cepstral coefficients in the matrix $\boldsymbol{X}'$, through the retained first p principal components of the covariance matrix $\boldsymbol{C}' = \widetilde{\boldsymbol{X}}'{\widetilde{\boldsymbol{X}}'}^T$. In this way, when considering the cepstral coefficients from the training dataset, the optimal output values in $\widehat{\boldsymbol{X}}'$ will result in a more stable probability distribution, representative of the undamaged state of the structure, compared to the distribution produced by the TAE (the results will be shown in Section 4.3). These derivations demonstrate again that the NGAE can better generalize the overall structural properties embedded in the cepstral coefficients by removing a large amount of the

variance of the cepstral coefficients contributed by the sensor and force locations, excitation and noise, resulting in a more robust training process with less risk of data overfitting.

4.2.2.4 Implementation of the proposed NGAE

In the field of deep learning, training a neural-network architecture commonly needs a series of ordinated steps and experiments to adjust the weights and biases of its neurons and layers [76]. The network hyperparameters, including both model (e.g., the numbers of layers, neuros, etc.) and algorithm (e.g., learning rate, mini-batch size, etc.) hyperparameters, can have significant effects on the network performance and are commonly determined by trial-and-error and by the rule-of-thumb [77]. Since cepstral coefficients can provide an effective and compact representation of the structural properties (natural frequencies, mode shapes, etc.), we employed a very concise autoencoder framework that consists of one input layer, one hidden layer, and one output layer for both the TAE and the proposed NGAE. The reason for choosing such a simplified architecture is that, theoretically, additional hidden layers in the TAE and NGAE architectures can improve the reconstruction capabilities but, at the same time, can also increase the risk of data overfitting and do not improve their capabilities of detecting damage from the variation patterns of the cepstral coefficients (as shown in Section 4.3). It is the handling of a reduced variance ground truth that enhances the damage identification capabilities of the NGAE, Therefore, such a simplified architecture can provide appreciable damage assessment performance with a robust learning process and, at the same time, significantly decrease the risk of data overfitting.

For a reasonable comparison between the results produced by the TAE and NGAE, almost identical sets of hyperparameters were used for both autoencoders (**Table 4.2**). The input and output layer sizes of the TAE and NGAE were both set equal to the number of the cepstral coefficients Q (Eq. (4.8)). For the size of hidden layer, it was set equal to rank $r_{\boldsymbol{X}'}$ of $\boldsymbol{X}'$ for the

NGAE, and equal to the rank $r_{\boldsymbol{X}_d}$ of $\boldsymbol{X}_d$ for the TAE, so to achieve the best-rank-$r_{\boldsymbol{X}'}/r_{\boldsymbol{X}_d}$ approximation, as shown in Section 4.2.2.3, and at the same time to avoid the data overfitting and unnecessary computational burden. A sigmoid function, $\Phi(x) = 1/(1 + e^{-x})$, was chosen as the activation function of the hidden layers (encoders) of the both autoencoders. Such a choice can be justified by the following reasons: 1) Its nonlinearity supports a more complex mapping function that typically enables a better input reconstruction for the autoencoders when compared to those that use a linear activation function, 2) the sigmoid function can largely retain the truncated-SVD-approximation mechanism for the single-hidden-layer autoencoders [74], so that the best-rank-$r_{\boldsymbol{X}'}/r_{\boldsymbol{X}_d}$ approximation can still be achieved to a considerable extent at the hidden layer. In addition, other settings need to be defined before training the model (as summarized in **Table 4.1**): 1) The Xavier Initialization strategy [78] was used for randomly initializing the weights and the biases of both autoencoders, 2) the batch size, a hyperparameter that defines the number of samples to work through before updating the weights and biases of the autoencoders, was chosen equal to 32, a reasonable default value [78], 3) the number of epochs defining the number of times that the optimizer will work through the entire training dataset, was set equal to 1000 so to make the autoencoders fully converge through the training process, and 4) an Adam optimizer [73] was chosen to implement the training for the NGAE and TAE.

One important advantage of the NGAE over the corresponding TAE is that the matrix of the desired output of the NGAE, $\boldsymbol{X}'$, is independent of the locations where the acceleration responses have been recorded and this enables the selection of a constant hidden layer size for the NGAE when modeling the response of different locations. On the contrary, the matrix of the desired output of the TAE, $\boldsymbol{X}_d$, depends on the target location, and thus the dimension of the hidden layer size needs to be varied as the target changes. Consequently, for the code implementation of the NGAE,

the architecture only needs to be set once, as a "class" or a "function", at the beginning and then just call it multiple times for modeling every sensor location in the system. By contrast, we need multiple different architectures of the TAE with different hidden layer sizes to model different recording locations, resulting in relatively larger coding workload and lower efficiency.

In terms of computational efficiency, the use of cepstral coefficients, in the order of 30∼50 coefficients, can largely decrease the structure complexity of the proposed NGAE, resulting in a significantly fast training process, compared to methods that use the recorded acceleration response (in the order of 10,000 data points) as input and output of autoencoders [66]. Similarly, during the inference process, since the cepstral coefficients can be extracted much more quickly than other features, e.g., AutoRegressive (AR) coefficients or natural frequencies [55, 67], the inference speed of the NGAE is almost instantaneous e.g., within a few seconds.

Table 4.1: The determined hyperparameters for the considered TAE and NGAE.

Property	Value
Input layer size	Q
Hidden layer size	$r_{\boldsymbol{X}_d}$ (TAE) or $r_{\boldsymbol{X}'}$ (NGAE)
Output layer size	Q
Activation function (hidden layer)	Sigmoid
Activation function (Output layer)	Identity
Epoch	1000
Batch size	32
Learning rate	1e-3
Loss function	Mean Squared Error (MSE)
Optimizer	Adam

Table 4.2: The workflow of the TAE or NGAE modeling.

Step 1: Given the training dataset $\{\boldsymbol{x}_{1,d}, \dots, \boldsymbol{x}_{N_{tr},d}\}_{d=1}^{N_d}$, create the matrix $\boldsymbol{X}_d$ for the TAE, or create both the matrices $\boldsymbol{X}_d$ and $\boldsymbol{X}'$ for the NGAE, as shown in Section 4.2.2.1-4.2.2.2.

Step 2: Set up the TAE or NGAE by incorporating the model hyperparameters shown in **Table 4.1**: For the TAE, the hidden layer size is set equal to the rank $r_{\boldsymbol{X}_d}$ of the matrix $\boldsymbol{X}_d$; for the NGAE, the size is set equal to the rank $r_{\boldsymbol{X}'}$ of the matrix $\boldsymbol{X}'$.

Step 3: Randomly initialize the weights and biases of the TAE or NGAE by the Xavier Initialization.

Step 4: Train the TAE or NGAE by the input number of epochs, or until convergence.

4.2.3 Evaluation metrics for damage measurement

After the training process is completed, the TAE or NGAE autoencoders have learned how to characterize the overall structural properties that are linked to the undamaged state of the system and that are embedded in the input cepstral coefficients. When a new set of cepstral coefficients $\{\boldsymbol{x}_{1,d}, \dots, \boldsymbol{x}_{N_{te},d}\}_{d=1}^{N_d}$ obtained from the same system but in an unknown (damaged or undamaged) state becomes available, with N_{te} being the number of the testing instances related to the d^{th} recording location, the previous undamaged-state information that the autoencoders have learned in the training phase can provide a reference for assessing the structural conditions in this unknown state. In this work, we adopted a strategy for assessing the presence of structural damage based on the data reconstruction error of the trained TAE or NGAE [66]: Two evaluation metrics, namely the Normalized Root Mean Square Error (NRMSE) and the Standard Deviation Ratio (SDR), have been used for the damage assessment task. The NRMSE aggregates the magnitude of the prediction errors for various data points into a single measure of predictive power, and can remove the effect of different error scales when modeling different recording locations. The SDR is an indicator based on the ratio of 2 standard deviations and can be an informative statistical representation of the signal reconstruction error: Conceptually, it can be considered as an extension of the signal-to-noise ratio in digital signal processing, originally developed to compare the level of a desired signal to the level of its background noise. In this work, we adopt the following expressions for computing the NRMSE and SDR values of each single instance in the training set $\{\boldsymbol{x}_{1,d}, \dots, \boldsymbol{x}_{N_{tr},d}\}_{d=1}^{N_d}$ or in the testing set $\{\boldsymbol{x}_{1,d}, \dots, \boldsymbol{x}_{N_{te},d}\}_{d=1}^{N_d}$:

$$NRMSE_{i,d}^{(tr)\ or\ (te)} = \frac{\sqrt{\frac{1}{Q}\left\|\boldsymbol{y}_{i,d} - \widehat{\boldsymbol{y}}_{i,d}\right\|_F^2}}{\max(\overline{\boldsymbol{x}}) - \min(\overline{\boldsymbol{x}})} \tag{4.24}$$

$$SDR_{i,d}^{(tr)\ or\ (te)} = \frac{\sigma_{\boldsymbol{y}_{i,d}}}{\sigma_{\widehat{\boldsymbol{y}}_{i,d}}} \tag{4.25}$$

where the vector $\boldsymbol{y}_{i,d}$ represents the i^{th} desired output $\boldsymbol{x}_{i,d}$ of the TAE or $\boldsymbol{x}_i'$ of the NGAE, and the vector $\widehat{\boldsymbol{y}}_{i,d}$ the i^{th} reconstructed output $\widehat{\boldsymbol{x}}_{i,d}$ of the TAE or $\widehat{\boldsymbol{x}}_{i,d}'$ of the NGAE, with the superscript "(tr)" and $i = 1, \ldots, N_{tr}$ linked to the training set, and "(te)" and $i = 1, \ldots, N_{te}$ for the testing set. The vector $\overline{\boldsymbol{x}}$ represents the element-wise average of all the instances in the training set, i.e., $\overline{\boldsymbol{x}} = \frac{1}{N_d}\frac{1}{N_{tr}}\sum_{d=1}^{N_d}\sum_{i=1}^{N_{tr}}\boldsymbol{x}_{i,d}$, and $\max(\overline{\boldsymbol{x}})$ and $\min(\overline{\boldsymbol{x}})$ represent the maximum and minimum values in the average vector $\overline{\boldsymbol{x}}$, respectively. The notations $\sigma_{\boldsymbol{y}_{i,d}}$ and $\sigma_{\widehat{\boldsymbol{y}}_{i,d}}$ are used to represent the standard deviations of the elements in the desired output $\boldsymbol{y}_{i,d}$ and in the reconstructed output $\widehat{\boldsymbol{y}}_{i,d}$, respectively.

4.2.4 Using the NGAE in a damage assessment strategy

Once the values of the two metrics (NRMSE and SDR) have been computed, an unsupervised statistical-pattern-recognition strategy is proposed for damage detection and quantification among different recording locations. The damage detection task can be conducted through the approach of multivariate novelty detection by establishing a statistical distribution, based on the training data, that is representative of the structural system in its undamaged state and by using the Squared Mahalanobis Distance (SMD) [79] on the output data from the training phase. A threshold on the SMD needs to be defined based on the training data distribution and, when a testing set from the structure in a new unknown state becomes available, the SMDs of the testing set will be compared

with the threshold to determine whether the system is in an undamaged state or not. After the occurrence of damage has been detected, the damage quantification for various damage scenarios can be implemented by defining and using proper SMD-based damage indices.

The results of the numerical study (discussed in Section 4.3) validated that the computed NRMSE follows a log-normal distribution, while the SDR follows a normal distribution. Therefore, the SMDs were computed on the ln(NRMSE), i.e., the natural logarithm of the NRMSE, and the original SDR, combined into 2-dimensional vectors $\boldsymbol{v}_{i,d}^{(tr)} \in \mathbb{R}^2 \ (i = 1, \dots, N_{tr})$ and $\boldsymbol{v}_{i,d}^{(te)} \in \mathbb{R}^2 \ (i = 1, \dots, N_{te})$ for the training and testing sets, respectively, as:

$$\boldsymbol{v}_{i,d}^{(tr)} = \left[\ln\left(NRMSE_{i,d}^{(tr)}\right),\ SDR_{i,d}^{(tr)}\right]^T \tag{4.26}$$

$$\boldsymbol{v}_{i,d}^{(te)} = \left[\ln\left(NRMSE_{i,d}^{(te)}\right),\ SDR_{i,d}^{(te)}\right]^T \tag{4.27}$$

The SMD of the vector $\boldsymbol{v}_{i,d}^{(tr)}$ can then be expressed as:

$$D^2(\boldsymbol{v}_{i,d}^{(tr)}) = \left(\boldsymbol{v}_{i,d}^{(tr)} - \boldsymbol{\mu}_d^{(tr)}\right)^T (\boldsymbol{\Sigma}_d^{(tr)})^{-1} (\boldsymbol{v}_{i,d}^{(tr)} - \boldsymbol{\mu}_d^{(tr)}) \tag{4.28}$$

where $\boldsymbol{\mu}_d^{(tr)} \in \mathbb{R}^2$ represents the sample mean over the N_{tr} instances of $\boldsymbol{v}_{i,d}^{(tr)}$, given by:

$$\boldsymbol{\mu}_d^{(tr)} = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \boldsymbol{v}_{i,d}^{(tr)} \tag{4.29}$$

and $\boldsymbol{\Sigma}_d^{(tr)} \in \mathbb{R}^{2\times 2}$ represents the covariance matrix, given by:

$$\boldsymbol{\Sigma}_d^{(tr)} = \frac{1}{(N_{tr} - 1)} \sum_{i=1}^{N_{tr}} (\boldsymbol{v}_{i,d}^{(tr)} - \boldsymbol{\mu}_d^{(tr)})(\boldsymbol{v}_{i,d}^{(tr)} - \boldsymbol{\mu}_d^{(tr)})^T \tag{4.30}$$

When a new set of the vectors $\boldsymbol{v}_{i,d}^{(te)} \ (i = 1, \dots, N_{te})$ from a testing dataset becomes available, the corresponding SMDs can be evaluated with respect to the established training distribution as:

$$D^2(\boldsymbol{v}_{i,d}^{(te)}) = \left(\boldsymbol{v}_{i,d}^{(te)} - \boldsymbol{\mu}_d^{(tr)}\right)^T (\boldsymbol{\Sigma}^{(tr)})^{-1} (\boldsymbol{v}_{i,d}^{(te)} - \boldsymbol{\mu}_d^{(tr)}) \tag{4.31}$$

A study by Ververidis et al. [80] has shown that the SMD of a n_o-dimensional multivariate data sample from a testing dataset, in this case $D^2(\boldsymbol{v}_{i,d}^{(te)})$, follows a scaled F-distribution with parameters of n_o and $N_{tr} - n_o$, where the distribution is defined by the N_{tr} instances of the training set. For our case, the SMD of $\boldsymbol{v}_{i,d}^{(te)}$ thus follows the scaled F-distribution that can be expressed by:

$$\frac{N_{tr}(N_{tr} - n_o)}{(N_{tr}^2 - 1)n_o} D^2\left(\boldsymbol{v}_{i,d}^{(te)}\right) \sim F_{n_o, N_{tr}-n_o}^{(d)} \tag{4.32}$$

where $n_o = 2$ for the two-dimensional vector $\boldsymbol{v}_{i,d}^{(te)}$. To detect the presence of damage, a threshold η_d, for the d^{th} recording location, was set equal to the 0.95-quantile of the training scaled $F_{n_o, N_{tr}-n_o}^{(d)}$ distribution. Consequently, the Scaled Squared Mahalanobis Distance (SSMD) of $\boldsymbol{v}_{i,d}^{(te)}$ was defined as a location-dependent damage index, given by:

$$SSMD\left(\boldsymbol{v}_{i,d}^{(te)}\right) = \frac{N_{tr}(N_{tr} - n_o)}{(N_{tr}^2 - 1)n_o} D^2\left(\boldsymbol{v}_{i,d}^{(te)}\right) \tag{4.33}$$

If the SSMD of the i^{th} instance $\boldsymbol{v}_{i,d}^{(te)}$ of the testing set is larger than the set threshold η_d, the system will be considered damaged over the corresponding i^{th} monitored event, and, if the median value of all the SSMDs of $\boldsymbol{v}_{i,d}^{(te)}$ for $i = 1, \dots, N_{te}$ is larger than η_d, the system will be considered damaged over the entire testing period.

Because the proposed damage-assessment strategy should be implemented for each recording location separately, there will be eventually N_d defined thresholds η_d $(d = 1, \dots, N_d)$ and thus can lead to some inconsistent conclusions. To provide a consistent scale for damage quantification, another damage index, termed as the Relative Scaled Squared Mahalanobis Distance (RSSMD),

was defined to remove the dependence of the damage index at a given location from its corresponding threshold, expressed as:

$$RSSMD_d = SSMD_{median,d} - \eta_d \tag{4.34}$$

where $SSMD_{median,d}$ represents the median value of the SSMDs of the testing instances $\boldsymbol{v}_{i,d}^{(te)}$ for $i = 1, \dots, N_{te}$ at the d^{th} location of the system: a positive value of the RSSMD thus indicates that the system has suffered some structural damage.

A flowchart of the entire damage assessment process is shown in **Figure 4.3**.

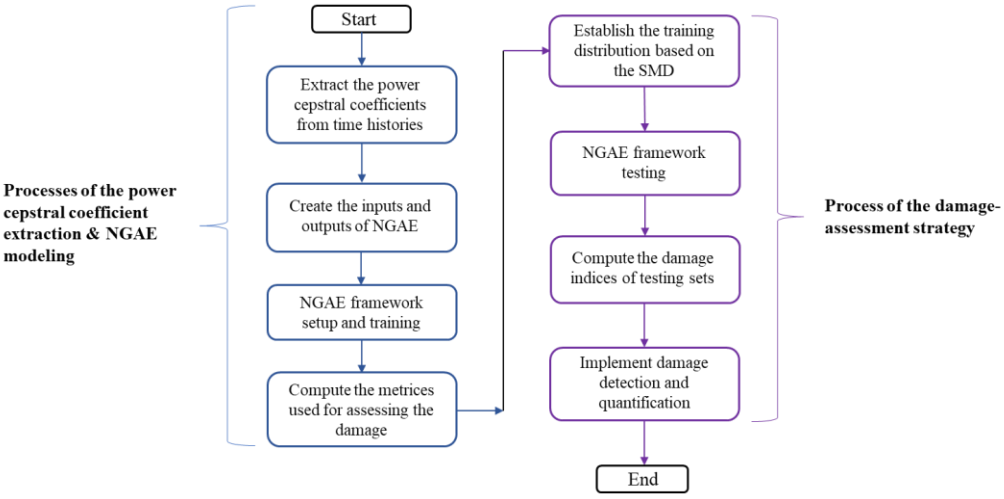

Figure 4.3: A flowchart of the proposed method for structural damage assessment

4.2.5 Computational requirements

The proposed NGAE architecture, integrated with the power cepstral coefficients, by virtues of its concise and easy-setup structure without deep hidden layers, can be used in rapid damage assessment tasks (Section 4.3) with few computational requirements. The TAE and NGAE architectures evaluated were run on a standard computer with Intel (R) core (TM) 3.89 GHz CPU and 16Gb of memory. The code was written in MATLAB (for the cepstral-coefficient extraction) and Python 3 (for the TAE and NGAE modeling). The CPU time for the whole process of the

coefficient extraction and 1000-epoch training is about 120-125 seconds for the numerical case study of the 8 DOF system (Section 4.3.1), and 45-50 seconds for the case study of the Z24 bridge (Section 4.3.2). Therefore, a standard machine with one CPU could easily provide the required computational power needed in real-life applications in structural damage assessment

4.3 Numerical studies and results

Two case studies were conducted to validate the effectiveness of the proposed method for structural damage assessment. In the first case, the cepstral coefficients are extracted from the simulated time-histories of the structural acceleration from an 8 DOF shear-type discrete model (Section 4.3.1) considering a variety of different damage scenarios, while, in the second case, the cepstral coefficients are obtained from the time-histories of the acceleration response recorded by a network of sensors installed on a real-bridge structure (Section 4.3.2) in progressive damage states.

4.3.1 Structural damage assessment of an 8DOF shear-type system

The first case study is represented by a lumped mass model of an 8 DOF shear-type system, as shown in **Figure 4.4**. The baseline conditions of the system are: The baseline stiffness of the vertical elements is set to $k_d^0 = 25{,}000\ \mathrm{N/m}\ (d = 1, \dots, 8)$, and each mass is equal to $m_d = 1\ kg\ (d = 1, \dots 8)$. The assumption of modal damping is used, with a damping factor of $\xi = 1\%$ for each of the 8 vibration modes.

To simulate different operational and damage conditions, sixteen different scenarios as shown in **Table 3** were considered by changing the baseline stiffnesses of certain elements. The first 9 scenarios represent the structural system in undamaged conditions, with only slight changes in the

stiffness at some floors, to simulate the fluctuations of the structural properties due to changing environmental conditions (e.g., temperature, humidity, etc.). The remaining 7 cases are representative of different structural damage conditions, with various types of drops in the stiffness of some vertical elements.

For each scenario, the excitation is provided by an external force applied either at the bottom mass (the 1st DOF) or at the top mass (the 8th DOF), with the probabilities of the force to act at DOF 1 or DOF 8 equal to 70% and 30% respectively, to simulate different statistical distributions and variances of the extracted cepstral coefficients. The external force is modeled as a zero-mean Gaussian white noise, with the zero-order-hold (ZOH) assumption and with a magnitude of 100 N. Each realization of the force has a duration of 500 seconds and it is sampled at 200 Hz. The generated acceleration time histories at the 8 DOFs are then corrupted by a 10% RMS Gaussian white noise to simulate measurement error. In this case study, 400 realizations of acceleration responses for each of the 9 undamaged scenarios shown in **Table 4.3** were simulated, for a total of 3600 sequences of the acceleration cepstral coefficients extracted at each DOF. All these data were then collected together to form the "training dataset" $\{\boldsymbol{x}_{1,d}, \dots, \boldsymbol{x}_{N_{tr},d}\}_{d=1}^{N_d}$ with $N_{tr} = 3600$ and $N_d = 8$. The "testing dataset" consists of the cepstral coefficients extracted from 200 new realizations of the acceleration responses for each of the 9 undamaged and of the 7 damaged scenarios, producing a testing set $\{\boldsymbol{x}_{1,d}, \dots, \boldsymbol{x}_{N_{te},d}\}_{d=1}^{N_d}$ with $N_{te} = 200$ and $N_d = 8$ for each of the 16 scenarios.

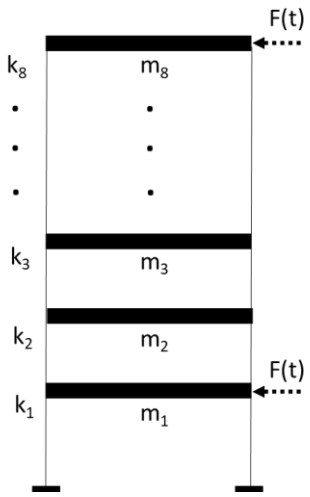

Figure 4.4: The 8 DOF shear-type system

Table 4.3: Considered undamaged and damaged scenarios of the 8 DOF shear-type system.

Scenario	Condition	Types of anomalies
1	Undamaged	Baseline scenario
2	Undamaged	$k_d = 0.98k_d^0$ for $d = 5,6,7,8$
3	Undamaged	$k_d = 0.99k_d^0$ for $d = 5,6,7,8$
4	Undamaged	$k_d = 1.01k_d^0$ for $d = 5,6,7,8$
5	Undamaged	$k_d = 1.02k_d^0$ for $d = 5,6,7,8$
6	Undamaged	$k_d = 0.98k_d^0$ for $d = 1,2,3,4$
7	Undamaged	$k_d = 0.99k_d^0$ for $d = 1,2,3,4$
8	Undamaged	$k_d = 1.01k_d^0$ for $d = 1,2,3,4$
9	Undamaged	$k_d = 1.02k_d^0$ for $d = 1,2,3,4$
10	Damaged	$k_d = 0.9k_d^0$ for $d = 1$
11	Damaged	$k_d = 0.9k_d^0$ for $d = 3$
12	Damaged	$k_d = 0.9k_d^0$ for $d = 5$
13	Damaged	$k_d = 0.9k_d^0$ for $d = 7$
14	Damaged	$k_d = 0.85k_d^0$ for $d = 7$
15	Damaged	$k_d = 0.9k_d^0$ for $d = 3,7$
16	Damaged	$k_d = 0.9k_d^0$ for $d = 2,8$

The TAE and the proposed NGAE were implemented following the scheme presented in **Table 4.2** and using a number of cepstral coefficients that ranged from 1 to 50, so as to check the sensitivity of the results to the number of coefficients considered. The evaluation metrics, namely the ln(NRMSE) and SDR, of the training and testing sets for each DOF were then computed and analyzed.

A series of normality tests, using the one-sample Kolmogorov–Smirnov (K-S) test [53], was then carried out over the estimated evaluation metrics and the corresponding p-values are shown in **Table 4.4** and **Table 4.5**. From the analysis of the results, it appears that the p-values related to the NGAE for both metrics are well above the 10% significance level ($\alpha = 0.1$), indicating that we can assume that both the ln(NRMSE) and the SDR follow a normal distribution. In addition, the p-values related to the NGAE are generally larger than the ones of the TAE, demonstrating that the NGAE, by substantially reducing the data variance attributed to the excitation and measurement noise, can establish a more robust 2-dimensional normal distribution as the training distribution for further damage detection and quantification.

Table 4.4: The p-values of the normality tests using the one-sample KS test for the ln(NRMSE) produced by the TAE and NGAE (the 8DOF shear-type case study).

	DOF 1	DOF 2	DOF 3	DOF 4	DOF 5	DOF 6	DOF 7	DOF 8
TAE	9.07E-3	2.97E-2	6.98E-2	5.14E-2	6.33E-2	5.54E-2	4.79E-2	2.65E-2
NGAE	4.63E-1	6.11E-1	6.72E-1	5.68E-1	6.24E-1	7.71E-1	6.03E-1	5.12E-1

Table 4.5: The p values of the normality tests using the one-sample KS test for the SDR produced by the TAE and NGAE (the 8DOF shear-type case study).

	DOF 1	DOF 2	DOF 3	DOF 4	DOF 5	DOF 6	DOF 7	DOF 8
TAE	2.47E-4	5.92E-2	4.22E-1	4.85E-1	5.14E-1	7.29E-2	3.48E-2	5.23E-3
NGAE	6.76E-1	7.55E-1	7.81E-1	8.98E-1	9.02E-1	7.68E-1	7.14E-1	6.06E-1

The scatter plots that visualize the distributions of the two metrics (ln(NRMSE) and SDR) are presented in **Figure 4.5**, where the results obtained from both the TAE and NGAE are provided for comparison. One can easily observe that the distributions of the two metrics from the undamaged scenarios in the testing set significantly overlap the ones of the training set, indicating that a well-established training distribution can be obtained by both the TAE and NGAE. With regard to the 7 damage scenarios (Scenario 10-16), the corresponding values of the two metrics

deviate from the training distribution to some degree due to different settings of damage severity, with the ones produced by the NGAE showing larger deviation patterns than these obtained by the TAE. This indicates that the NGAE can better characterize the damage characteristics embedded in the cepstral coefficients than the TAE.

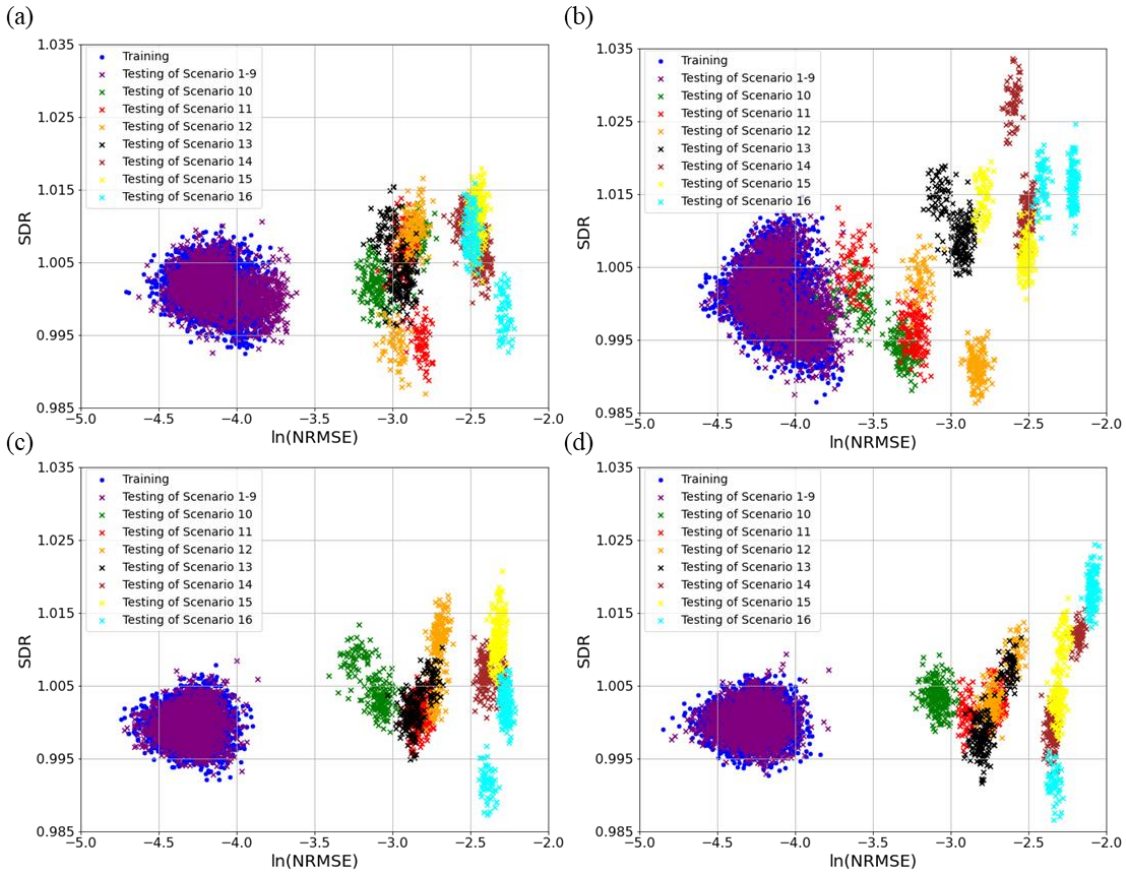

Figure 4.5: The distributions of the ln(NRMSE) and SDR produced by the TAE and NGAE for the 9 undamaged scenarios and the 7 damage scenarios, considering 50 cepstral coefficients (Q = 50). (a) The results of the TAE at the 3rd DOF. (b) The results of the TAE at the 7th DOF. (c) The results of the NGAE at the 3rd DOF. (d) The results of the NGAE at the 7th DOF.

The damage detection for this case study was conducted for each of the 8 DOF, based on the 8 thresholds η_d $(d = 1, \dots, 8)$ obtained from the established training distributions. Then, the SSMDs of $\left\{\boldsymbol{v}_{i,d}^{(te)}\right\}_i^{N_{te}}$ $(d = 1, \dots, 8$ and $N_{te} = 200)$, related to each of the 16 scenarios, were computed and individually compared with the corresponding η_d for a binary classification,

assigning ’0’ if the system was classified as undamaged or ‘1’ if damaged. The confusion matrices in **Figure 4.6** show the classification results of the TAE and NGAE, at the 3rd and 7th DOFs, for all testing scenarios (200 × 16 instances), corresponding to the 5% significance level (Section 4.2.4). The overall accuracies of both TAE and NGAE are excellent, with the NGAE performing slightly better than the TAE (e.g., false positive rate 0.53% – 0.66% vs. 1.94% – 2.19%). It is important to note that the classification results of both TAE and NGAE can provide extremely small Type-II errors (i.e., the damaged scenarios are misclassified as undamaged), with very low error rates (< 0.38% for the TAE and basically 0 for the NGAE). In addition, by comparing these results with those presented in [1], the proposed NGAE can also outperforms the PCA method, with the latter one leading to relatively larger Type-II errors (an F1 score of 96.8% (PCA) vs. over 99.3% (NGAE)). The receiver operating characteristic (ROC) curves, as the overall classification performance of the TAE and NGAE, are shown in **Figure 4.7**, where the NGAE can provide a larger area under the curve (AUC) compared to the TAE (0.999 vs. 0.996), demonstrating the superiority of the NGAE over the TAE in the damage detection again.

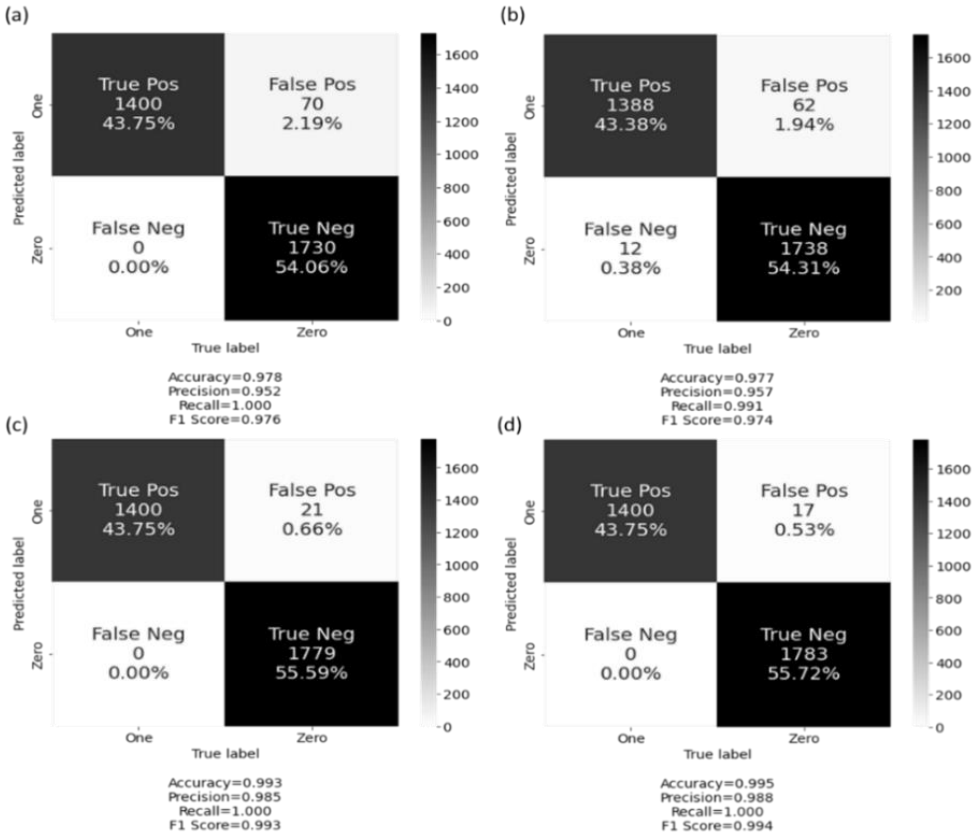

Figure 4.6: Confusion matrices of the binary classification at the 3rd and 7th DOF of the 8 DOF system, for both TAE and NGAE (Q = 50), corresponding to the 5% significance level. (a) The results of the TAE at the 3rd DOF. (b) The results of the TAE at the 7th DOF. (c) The results of the NGAE at the 3rd DOF. (d) The results of the NGAE at the 7th DOF.

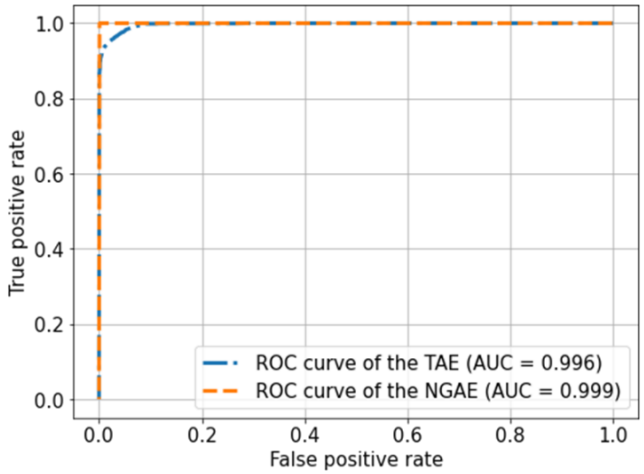

Figure 4.7: ROC curves of the binary classification performance of the TAE and NGAE, produced by averaging the results of the 8 DOFs.

Since the number of the cepstral coefficients used in the analysis (Q) is linked to the input and output dimension of the TAE and NGAE, an investigation was conducted to explore the relationship between the number Q and the damage-detection accuracy, by recording the trends of the F-1 scores over varying Q as shown in **Figure 4.8 (a)**. The results show that, for $Q < 30$, the F-1 scores rise quite rapidly and then, as Q increases, they stabilize approaching 100. This trend is related to the fact that the magnitude of the cepstral coefficients decreases with increasing quefrency. For lower Q, more and more information on the structural properties become available to the autoencoders as Q increases but, for large Q, little to no new information is acquired. This can also be seen by looking at the variation of the rank r of the cepstral coefficient matrices $\boldsymbol{X}_d$ ($d = 1, \ldots, 8$) and $\boldsymbol{X}'$ as a function of Q (**Figure 4.8 (b)**): It is obvious that beyond a certain range of Q, adding more cepstral coefficients cannot further improve the performance of the autoencoder.

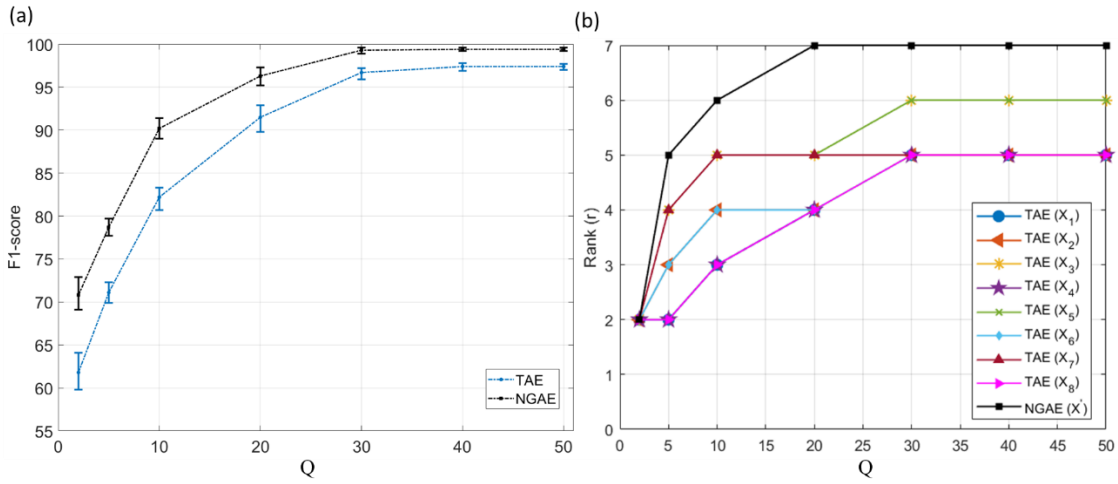

Figure 4.8: (a) The average F1-score of the 8 DOF over Q. (b) The ranks of the matrices X_d (d = 1, …, 8) and X' over Q. The error bars in (a) represent the minimum and maximum F1-scores of the 8 DOF's results.

After obtaining the values of the damage index SSMDs for the 16 testing sets, a further step for damage quantification was conducted by computing the RSSMDs of the 16 testing sets (Eq. (4.34)) at each DOF. As a result, the RSSMD of the 8 DOF under the 16 scenarios, produced by

the TAE and NGAE, are shown in the **Figure 4.9** and **Figure 4.10**, respectively. Looking at these plots, the following observations can be made: 1) The RSSMDs of all the undamaged scenarios 1-9 are negative and the ones of the damaged scenarios 10-16 are positive. Such results are in perfect agreement with the definition of the RSSMD (Eq. (4.34)), proving that both the proposed NGAE and the TAE can accurately identify the damage and undamaged conditions. 2) For the damaged scenario 10-16, the magnitude of the RSSMDs produced by the TAE and NGAE is directly related to the damage severity, showing that the two autoencoder architectures can provide close results in indicating occurrence and severity of damage. 3) The RSSMDs produced by the NGAE for the different damage conditions are generally larger than the ones obtained by the TAE: this confirms that the proposed NGAE, by using the weighted average as the output, better characterizes the overall structural properties embedded in the cepstral coefficients and shows great sensitivity in assessing structural damage.

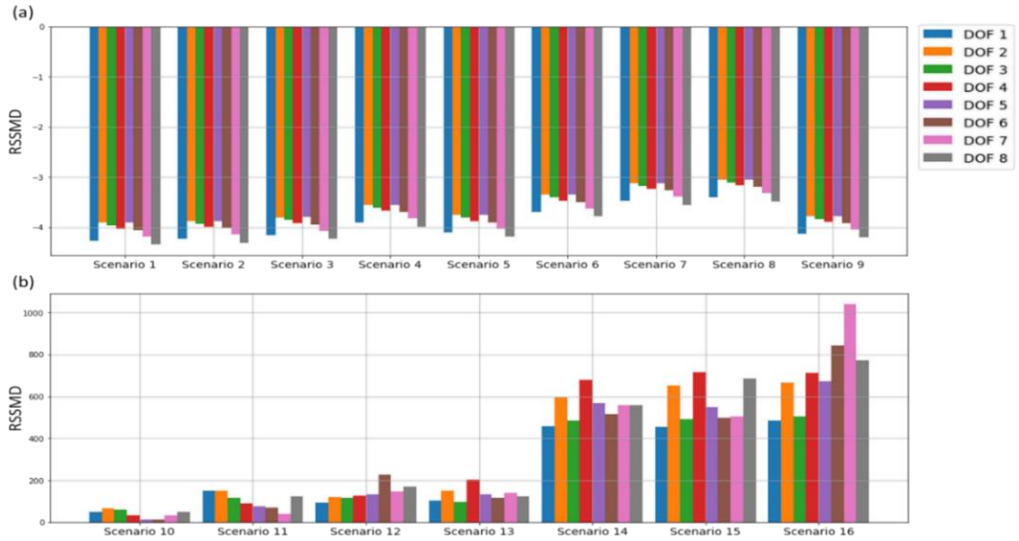

Figure 4.9: The RSSMDs of the 8 DOF across the 16 scenarios, produced by the TAE (Q = 50). (a) The results of the 9 undamaged scenarios. (b) The results of the 7 damaged scenarios.

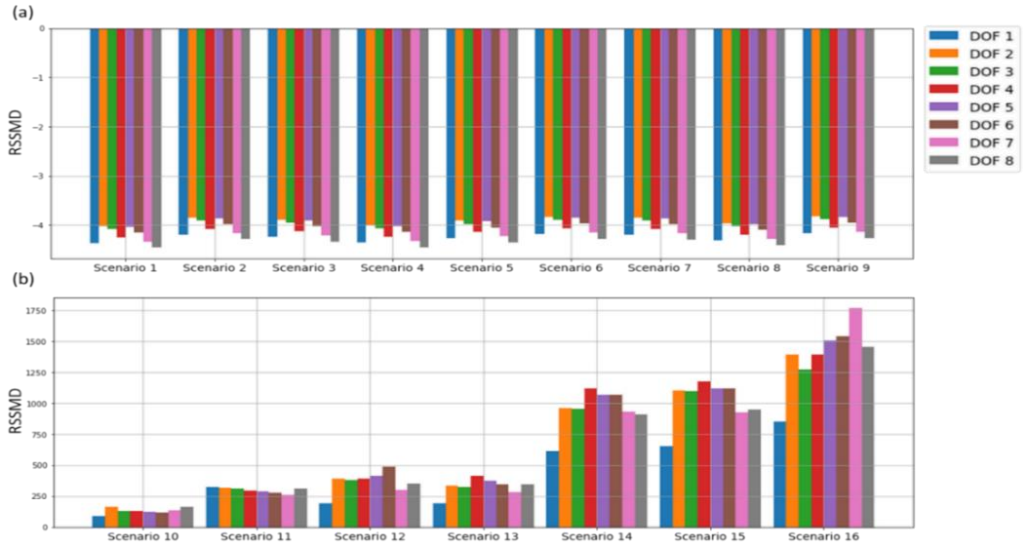

Figure 4.10: The RSSMDs of the 8 DOF across the 16 scenarios, produced by the NGAE (Q = 50). (a) The results of the 9 undamaged scenarios. (b) The results of the 7 damaged scenarios.

4.3.2 Structural damage assessment of the Z24 bridge

The data recorded during operation and demolition of the Z24 bridge, a well-known case study, were used to evaluate the performance of the proposed NGAE in dealing with data from real applications. The Z24 bridge was a concrete box girder bridge, with a main span of 30 m and two side spans of 14 m, in the canton of Bern, Switzerland. This bridge was monitored for about 10 months (Nov. 10, 1997 – Sep. 10, 1998), with the intent to analyze the effects of some environmental parameters such as local temperature, rain, wind speed, humidity, traffic condition, etc., on the structural response. At the end of the 10-month period and prior to its final demolition, progressive damage in terms of lowering of a pier, spalling of concrete, etc., was induced on the bridge and the corresponding responses were recorded. **Table 4.6** gives an overview of the various monitoring campaigns and damage conditions of the bridge. More detailed information about the bridge and its monitoring system can be found in Kramer et al. [81] and Reynders et al. [82].

For each damage condition, the bridge was subjected to an ambient vibration test and to a forced vibration test, with two vertical shakers placed on the bridge deck to provide the forced

excitation with a smooth and stable spectrum between 3 and 30 Hz. A network of 16 accelerometers, positioned at strategic location on the bridge (**Figure 4.11**), was set up to record structural acceleration responses. For every hour, a total of 65,536 samples (with a sampling time interval of 0.01 s) were recorded by each accelerometer, using an antialiasing filter with a cutoff frequency of 30 Hz.

Table 4.6: An overview of the various bridge structural conditions.

Date (1998)	Scenario
10-17 July	**Undamaged condition**
4 August	**Undamaged condition**
9 August	Installation of pier settlement system
10 August	Lowering of pier, 20 mm
12 August	Lowering of pier, 40 mm
17 August	Lowering of pier, 80 mm
18 August	Lowering of pier, 95 mm
19 August	Lifting of pier, tilt of foundation
20 August	New reference condition
25 August	**Spalling of concrete at soffit, 12 m^2**
26 August	**Spalling of concrete at soffit, 24 m^2**
27 August	Landslide of 1 m at abutment
31 August	**Failure of concrete hinge**
2 September	**Failure of 2 anchor heads**
3 September	**Failure of 4 anchor heads**

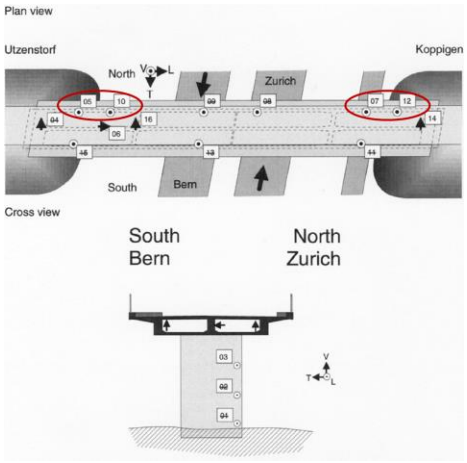

Figure 4.11: Details of the locations of the setup sensors in Z24 bridge. The considered accelerometers 05, 07, 10, and 12 are circled by the 2 red circles.

In this case study, the damage assessment operation was carried out using only the data recorded by the accelerometers 05, 07, 10, 12 (**Figure 4.11**), as the records of other accelerometers

were presented some abnormalities [55]. The recorded acceleration responses of the first two scenarios on **Table 4.6** (July 10^{th} – 17^{th} and August 4^{th} – 9^{th}), representative of the undamaged conditions, were used for training the proposed NGAE: Although the environmental conditions (e.g. temperature, humidity, wind) were quite similar, the data recorded between July 10^{th} – 17^{th} were obtained from the bridge in its regular operational conditions while those from August 4^{th} – 9^{th} correspond to forced vibration tests. The data corresponding to the other 5 scenarios were used for testing: The data from sensor 10 during the August 27^{th} – 31^{st} period were not available and so it was not considered. In order to increase the dataset size with more instances, each of the hourly records was framed into three 30 minutes segments, with 15 minutes overlapping. Accordingly, a total of 684 framed records of the two undamaged scenarios, for each of the 4 sensors, were available. The training dataset was created by randomly selecting 90% of the data for each of these two scenarios ($N_{tr} = 616$), while the data were used as 2 undamaged testing sets. Similarly, the recorded acceleration responses of the 5 considered damaged scenarios were preprocessed in a similar fashion resulting in 5 damaged testing sets. In total, there were 7 testing datasets for a total of 245 available framed records.

Similarly to the numerical study, 50 cepstral coefficients $c_{i,d}[q]$, $q = 1, \dots, 50$ ($Q = 50$), were extracted from each record, with the first one of each sequence ($q = 0$) discarded. When dealing with either the training or the testing datasets, the vector $\boldsymbol{x}_i'$ (defined by Eq. (4.12)) is the weighted summation of the vectors $\boldsymbol{x}_{i,j}$ for $j = 1, 2, 3, 4$, linked to the sensors 05, 07, 10, 12, respectively. The NGAE and TAE were set up and trained by using the same hyperparameters and strategy shown in **Table 4.1** and **Table 4.2**.

Figure 4.12 shows the distributions of the computed ln(NRMSE) and SDR for the data from sensor 12 obtained using the proposed NGAE, while **Table 4.7** and **Table 4.8** present the

comparisons of the corresponding p-values for both the TAE and the proposed NGAE. Looking at the p-values, one can easily observe that the p-values related to the NGAE are quite larger than those obtained by the TAE, and evident bias exists in the SDR of the data obtained by the TAE, probably due to overfitting issues caused by the large variance of the data linked to the external excitation and to the measurement noise. On the contrary, the proposed NGAE, by working with weighted averages of all the cepstral coefficients, can provide a more stable 2-dimensional normal distribution for the training data. The ability of the proposed NGAE in differentiating data corresponding to undamaged conditions from those of damaged conditions is visualized in **Figure 4.12**, which shows the distributions of the ln(NRMSE) and SDR obtained by the NGAE. It is clear that the portion of test data corresponding to the undamaged condition perfectly fits with the training distribution (**Figure 4.12 (a)**) while, when damage occurs, the data clearly deviate from it (**Figure 4.12 (b)**).

Table 4.7: The p values of the normality tests using the one-sample K-S test for the ln(NRMSE) produced by the TAE and NGAE (the Z24 bridge case study).

	Sensor 5	Sensor 7	Sensor 10	Sensor 12
TAE	6.49E-3	1.62E-2	8.06E-4	9.55E-3
NGAE	1.15E-1	2.77E-1	2.42E-1	1.08E-1

Table 4.8: The p values of the normality tests using the one-sample K-S test for the SDR produced by the TAE and NGAE (the Z24 bridge case study).

	Sensor 5	Sensor 7	Sensor 10	Sensor 12
TAE	4.03E-3	7.29E-3	3.18E-6	9.88E-4
NGAE	1.29E-1	4.98E-1	4.45E-1	3.51E-1

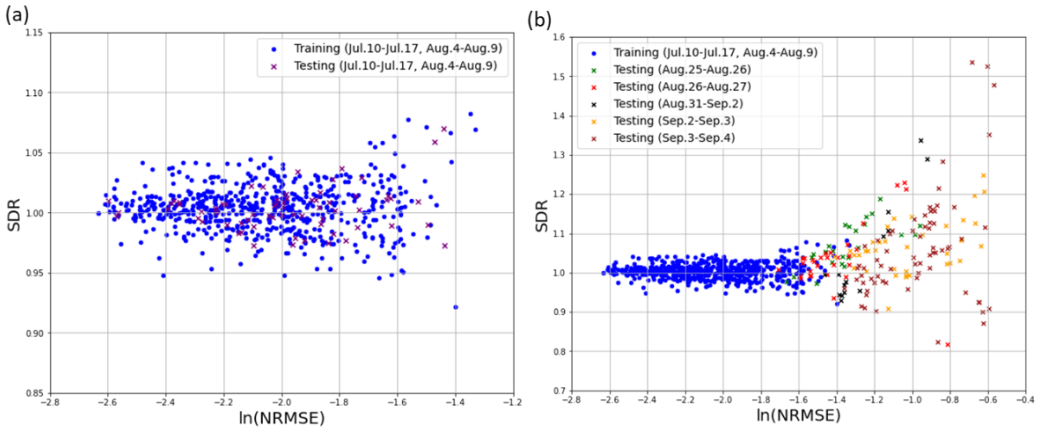

Figure 4.12: The distributions of the ln(NRMSE) and SDR obtained from sensor 12 using the NGAE. (a) presents the results corresponding to the training data and undamaged testing data. (b) presents the results corresponding to the training data and damaged testing data.

The damage detection and quantification operations were conducted, following the proposed statistical-pattern-recognition strategy (Section 4.2.4). First, the damage index SSMDs (Eq. (4.33)) were computed for the data corresponding to each of the 4 sensors separately. Next, for the computed SSMD values of the 7 testing sets obtained from each of the 4 sensors, they were individually compared with the threshold value set for the corresponding sensor (Eq. (4.32) - (4.33)). The confusion matrices in **Figure 4.13** present the classification accuracies obtained by the proposed NGAE for the data from sensor 10 and 12. It can be seen that the proposed NGAE is still quite accurate in detecting damage even when dealing with real data. Here, it is noted that the type II error in the classification is slightly larger than the type I error, mainly due to the low damage level of the bridge during August 25th – 27th monitoring campaign, resulting in some corresponding SSMD values being close to those of the undamaged scenarios and thus being incorrectly classified as undamaged. In addition, the median value of the SSMDs of each damage scenario was compared with the threshold value set for each sensor to determine whether the bridge was damaged or not. **Figure 4.14** shows the boxplots of the distributions relative to the data from the sensor 12 for the 7 scenarios, obtained by the proposed NGAE: The value of the threshold has

been set equal to 4.39 (dash blue line in **Figure 4.14**) corresponding to the 5% significance level (Section 4.2.4). By looking at these distributions, the ones corresponding to the undamaged cases are well below the threshold while, when damage is present, the distributions shift above the threshold. From these results, it is evident that the proposed NGAE can accurately classify damaged and undamaged scenarios.

In order to quantify the different levels of damage severity, the damage indices RSSMDs (Eq. (4.34)) for each of the 4 recording locations were computed, and the results are shown in **Figure 4.15**. These results confirm that the RSSMD values obtained at the 4 sensor locations can generally provide accurate assessment for the undamaged and damaged conditions. For low damage levels as the ones on Aug 25^{th} – 27^{th}, the RSSMD values from sensors 05 and 07 are quite small and negative while those from sensor 10 and 12 are still small but positive. This difference in sign is due to the fact that, for low levels of damage, the deviation of the testing data from the undamaged training distribution is quite small and this could result in different signs of the RSSMD values. It is then recommended that the results from multiple locations be considered simultaneously in a damage assessment strategy. For more severe damage conditions, like the failure of a concrete hinge or the failure of anchor heads, the RSSMD values are significantly higher than the ones of the first two damage scenarios, indicating that the bridge was under more serious cumulative damaged conditions over the period of Aug. 31^{st} – Sep. 4^{th}.

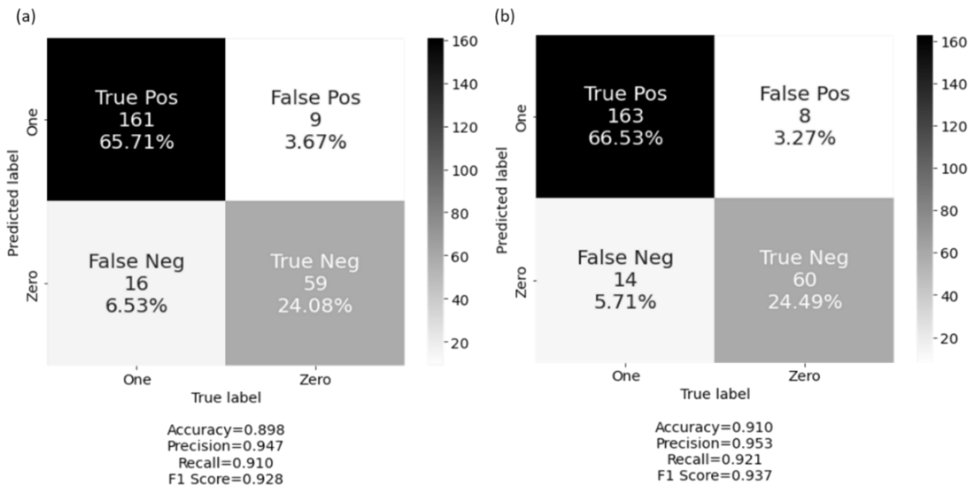

Figure 4.13: Confusion matrices of the binary classification for the Z24 bridge, obtained by the NGAE (Q = 50). (a) The results from sensor 10. (b) The results from sensor 12.

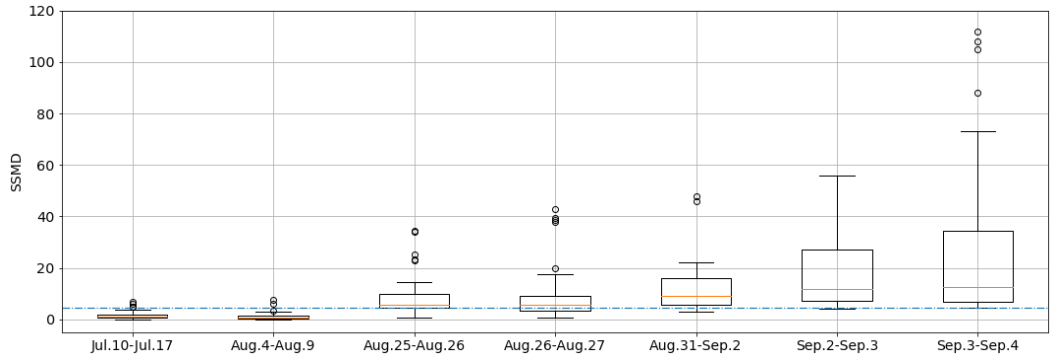

Figure 4.14: The distribution of the SSMDs for the 7 considered scenarios, with respect to the sensor 12, obtained by the NGAE. The dash blue line represents the defined threshold linked to the sensor 12, which is estimated equal to 4.39.

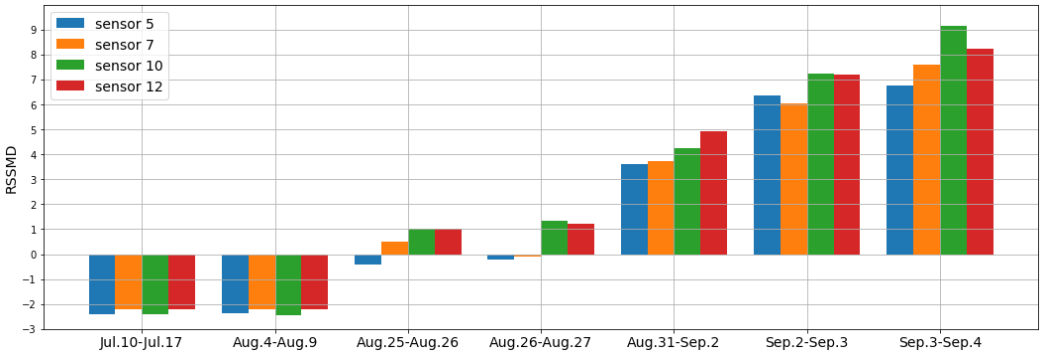

Figure 4.15: The RSSMDs of the 4 sensors for the 7 considered undamaged and damaged scenarios, obtained by the NGAE.

4.4 Conclusions

In this chapter, a New Generalized Autoencoder (NGAE) architecture, integrated with a statistical-pattern-recognition-based approach that uses power cepstral coefficients as the Damage Sensitive Features (DSFs), is proposed for structural damage assessment. The NGAE can be well-generalized in terms of the component of the cepstral coefficients that represent the structural properties of the overall system thanks to a newly defined encoder-decoder mapping. To validate the proposed NGAE, two case studies have been presented, namely an 8 DOF system excited by an external force and the benchmark problem of the Z24 bridge in Switzerland, with various undamaged and damaged scenarios. From the analysis of the results, it can be concluded that the NGAE can successfully characterize the overall structural properties embedded in the cepstral coefficients by virtues of the newly defined encoder-decoder mapping, largely reducing the effects of the variance attributed to the external excitation and to the measurement noise. This effect will result in an appreciable accuracy in the assessment of damage within the structural system. The following are the main conclusions drawn from this chapter:

1) Using the power cepstral coefficients as the inputs and outputs of the autoencoders, benefiting from their compact and effective representation of the structural modal properties, supports an efficient and robust damage-assessment strategy by significantly decreasing the network complexity. This leads to a significant reduction in overfitting the data and in the required computational resources, in comparison to methods that use the recorded acceleration responses or other traditional features (e.g., natural frequencies, mode shapes, etc.) as the inputs and outputs of autoencoders.

2) The proposed NGAE has an important advantage with respect to the TAE in terms of implementation: When modeling multiple recording locations of a system separately, the setting

for the hidden layer size of the NGAE is fixed, as the desired output of the NGAE is defined to be consistent across different sensor locations. On the contrary, the optimal setting for the hidden layer size of the TAE needs to change according to the desired output, leading to a relatively large workload of coding implementation.

3) For the case study of the 8 DOF shear-type system, the values of the two considered evaluation metrics (NRMSE and SDR) computed based on the NGAE are able to establish a more robust training distribution (supported by the results of the Kolmogorov–Smirnov (K-S) tests), leading to a higher damage detection accuracy compared to the traditional autoencoder (TAE) and the PCA.

4) In the Z24 bridge case study, the NGAE considerably outperforms the TAE, successfully detecting the presence of the damage and quantifying the damage severity for various structural conditions.

Acknowledgement

The authors gratefully acknowledge the Brite EuRam Programme BE-3157 SIMCES, the European Commission and the Structural Mechanics Section of KU Leuven for gathering and sharing the data of the Z24 bridge.

Chapter 5. A Data Augmentation Strategy for Structural Damage Classification

5.1 Introduction

As discussed in Section 4.1, the explosive development of Machine Learning (ML) methods in the last decade has led to a large amount of research efforts focusing on the application of these techniques in structural health monitoring. Among these methods, damage assessment strategies based on supervised learning have been proved to be effective in identifying different damage types and severity levels in civil structures [62]. However, although supervised strategies can provide fairly accurate damage assessment results, they need a proper and systematic model training process that generally requires large datasets representative of both the undamaged structure and the structure in different damage conditions, a requirement that cannot be obviously satisfied when dealing with real-life civil structures (i.e., buildings and bridges) [66].

For this type of structures, there is an abundance of data from the undamaged condition but only a few data from the structures in the presence of damage. To properly train a model, the data from other damaged structures need to be included but this cannot be easily done when dealing with vibration data. For example, two bridges with similar structural properties but different soil conditions could have a substantially different dynamic behavior. Therefore, in ML applications to civil structures, many recent studies on damage assessment have turned their attention to the development of unsupervised learning approaches, with the vast majority of data coming from the structure in its undamaged condition. For example, the developed New Generalized Auto-Encoder (NGAE) presented in Chapter 4 has been validated as an excellent unsupervised-learning method to solve such problems.

Nevertheless, even when it is possible to bypass the real-life problem represented by the lack of training data from the structure in damaged conditions, unsupervised-learning strategies show poor performance when classifying different types of structural damage since it is challenging for these strategies to self-discover distinguishable hidden patterns in unlabeled data for classification [83]. In recent years, in order to deal with the paucity of data from damaged structural conditions, researchers have been exploring the strategy of Transfer Learning (TL) [84] from a rich and large "source" domain to a "target" domain representative of civil structural systems: The idea behind TL is that a numerical/statistical model can be trained to gain the ability to catch changes in a signal from a domain with a rich dataset, and then transfer this knowledge to a signal from a somehow related "target" domain with much fewer training data, e.g., a building or a bridge, in order to better identify its structural conditions [85].

Along with the development of TL strategies in SHM applications is the exploration of a different strategy, called "data augmentation" [86], to deal with the data deficiency problem in structural damage conditions. Widely used in computer vision and natural language processing [87, 86], data augmentation, by increasing the size and improving the quality of the training dataset, is considered an effective solution to the problem of limited datasets, allowing the identification of better ML models. In SHM applications, Zhai et al. [88] used a 3D graphics model to generate synthetic data for augmenting a real-world image dataset of crack bridge girders. The augmented image data were then used to train a convolutional neural network for identifying fatigue cracks in steel structures. Wan et al. [89] developed a data augmentation technique to generate new samples of bridge monitoring data such as traffic flow, temperature, and strain, based on an improved architecture of Generative Adversarial Networks (GANs). Through an experimental study on data collected from a real bridge, their results showed that the proposed

strategy was successful in augmenting the original dataset and consequently improving the performance of both traditional and neural-network classifiers in evaluating the bridge's condition. It is within this framework that the work contained in this chapter finds its perfect fit.

The focus of this chapter is to introduce our efforts in developing an ML strategy that can be used for structural damage assessment in cases where only few data are available from the structure in damaged conditions. In buildings, bridges, dams, etc., there is a large amount of recorded data available: These data are mainly in the form of time-histories of the structural response (accelerations and/or displacements) to some external/ambient excitations. The vast majority of these data are obtained from the structure in its operational (or undamaged) condition and so such a dataset can be used to train an automated algorithm in recognizing the structural characteristics in the structure's operational state. When damage occurs, only a few data records are readily available and thus a ML algorithm can only perform an anomaly detection operation. To train an algorithm to classify different structural conditions (e.g., undamaged, small damage level, etc.), it is necessary to train an algorithm on a balanced dataset, where different damage classes have a roughly equal number of data samples.

To achieve this objective, we develop a novel data augmentation strategy based on a Conditional Variational Autoencoder (CVAE) architecture [90]. Once this CVAE-based model has been properly trained, can be used to generate new samples of a type of DSFs which augment the originally unbalanced dataset. The power cepstral coefficients of the recorded structural acceleration (Section 4.2.1) will represent the dataset of the Damage Sensitive Features (DSFs) that will be augmented. A new type of the power cepstral coefficients is considered, which can largely boost the performance and robustness of the data augmentation and consequently of subsequent structural damage classification task. The robust extraction process and the stable

statistical distribution of these cepstral coefficients support the approach of building appropriate probabilistic recognition models to describe them effectively. The proposed CVAE can use several conditional independent Gaussian distributions simultaneously, to model the distributions of the power cepstral coefficients obtained in various structural damage conditions in the latent space of the CVAE, with the help of newly defined conditional random variables. The conditional random variable of the CVAE, which is traditionally considered as the class label of a target dataset for augmentation, is defined in this work by an unsupervised-learning approach for addressing the prior unknown structural conditions.

The augmented dataset of the cepstral coefficients can then be employed to better train a Probabilistic Linear Discriminant Analysis (PLDA) [91] model for greater accuracy in damage classification. To handle the practical case of continuously updating the dataset with data coming from the structural system, a sliding-window strategy to timely update the classification model is proposed, with the corresponding results of a real bridge structure presented in this chapter.

5.2 Methodology

5.2.1 Cepstral coefficients of acceleration response as damage sensitive features

The performance of the proposed data augmentation strategy is investigated using two types of cepstral coefficients of the structural acceleration response, i.e., 1) the original cepstral coefficients, which have been presented in Section 4.2.1, and 2) a new type of weighted cepstral coefficients.

As a reminder, when extracting the original cepstral coefficients from a set of different time histories of the structural response recorded on a structural system in identical conditions (e.g., in the undamaged condition), the majority of the variance of the cepstral coefficients is attributed to

the term $\gamma_d[q]$ in Eq. (4.6), while the contribution from the term $\theta[q]$ in Eq. (4.6) should remain approximately constant for the various locations on the system, except for some inevitable measurement noise. Such a variation can mislead a model built for structural damage classification and prevent it from effectively characterizing the overall structural properties embedded in the cepstral coefficients. To take into account such a variation of the cepstral coefficients, we consider to employ a new type of the cepstral coefficients, which is originally defined as the desired output of the NGAE (as introduced in Section 4.2.2). According to its definition in Eqs. (4.12)-(4.15), this new type of the cepstral coefficients is a specific weighted summation of the cepstral coefficients from all recording locations of the system, and thus they are named as "weighted" cepstral coefficients in this chapter.

Two important points are noteworthy here: First, the weighted cepstral coefficients in $\boldsymbol{x}_i'$, defined in Eq. (4.15), are independent of the locations where the structural acceleration has been recorded. Hence, if we have a few sets of the original cepstral coefficients in $\boldsymbol{x}_{i,d}$, defined in Eq. (4.8), from all the recording locations $d = 1, \dots, N_d$, they will only produce one set of weighted cepstral coefficients in $\boldsymbol{x}_i'$. Second, the weighted summation $\sum_{j=1}^{N_d} s_j \gamma_{i,j}\,[q]$ in Eq. (4.15) can help shrink the data variance associated with the excitation and measurement location terms, indirectly enhancing the contribution of the term $\theta_i[q]$ that is linked only to the overall structural properties. Consequently, the weighted cepstral coefficients provide a more stable statistical distribution, i.e., a Gaussian distribution as demonstrated in Section 4.3, and thus can be more easily characterized and fitted by a probabilistic generative model, compared to the original cepstral coefficients.

In this work, we investigate two methods of that use both the original cepstral coefficients in $\boldsymbol{x}_{i,d}$ and the weighted coefficients in $\boldsymbol{x}_i'$, respectively, for the next steps of data augmentation and damage classification, and compare the results produced by both (Section 5.3).

5.2.2 Overview of variational autoencoders

The Variational Auto-Encoder (VAE) is a type of deep generative models that are aimed to simulate how the observed data are generated in the real world [92]; essentially, they are neural networks with multiple hidden layers that are trained to approximate probability distributions of observed data samples. The mechanism of VAEs can be interpreted as an integration of probabilistic graphical models [93] and deep learning techniques. The fundamental structure of a VAE consists of two dual parametric inference models, termed as the probabilistic encoder and the probabilistic decoder, with a latent space between the two for sampling a latent variable. During the training process of a VAE, the encoder is forced to learn a multivariate latent distribution that approximates the posterior distribution of its input variable. A sampling operation is then performed based on the approximate posterior distribution to generate latent variable samples, which are then passed to the decoder with the aim to reconstruct the input variable as its output. Once the training is completed, the trained decoder of the VAE can be then employed as a generative model to augment a limited dataset.

5.2.3. Conditional variational autoencoders

5.2.3.1 Motivation

A key disadvantage of using the decoder of the VAE as a generative model is that there is no control over what kind of data will be generated, since it is difficult to define during the sampling operation which part of the latent space of a trained VAE corresponds to the exact type of data to be generated. This can be problematic in damage classification problems when we generate data that are representative of a particular structural damage scenario, since it would be challenging to define explicit boundaries for different classes of data in the learned latent space of the VAE.

Therefore, we develop a data-augmentation strategy based on the Conditional VAE (CVAE) architecture [90] to control generating specific data that correspond to particular structural damage scenarios. **Figure 5.1** presents the fundamental mechanisms of the VAE and of the CVAE to intuitively visualize the difference between these two network architectures.

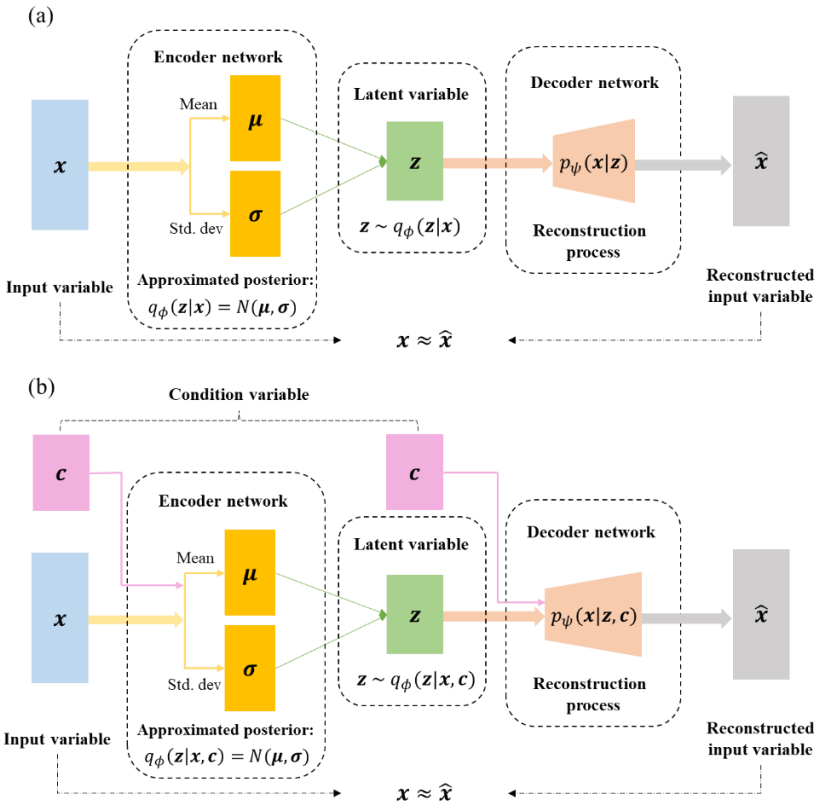

Figure 5.1: The fundamental mechanisms of the VAE (a) and CVAE (b).

5.2.3.2 The mechanism of the conditional variational autoencoder

To introduce the concept of the CVAE used in this work, let us consider a dataset $\{\boldsymbol{x}_i\}_{i=1}^{N}$ that accounts for N independent and identically distributed (i.i.d.) samples of the observed variable $x \in R^{Q}$, where $\boldsymbol{x}$, in this case, is a vector containing the values of Q cepstral coefficients (the original ones in Eq. (4.8) or the weighted ones in Eq. (4.15)). A probabilistic framework of the CVAE can be established by assuming that the dataset is generated through a random process that involves a latent variable $\boldsymbol{z}$ and a condition variable $\boldsymbol{c}$. For a given condition $\boldsymbol{c}$, the vector $\boldsymbol{z}$ is drawn from

the conditional prior distribution $p_\psi(\boldsymbol{z}|\boldsymbol{c})$, and $\boldsymbol{x}$ is obtained from the conditional distribution $p_\psi(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c})$, expressed as:

$$\begin{aligned} \boldsymbol{z} &\sim p_\psi(\boldsymbol{z}|\boldsymbol{c}) \\ \boldsymbol{x} &\sim p_\psi(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c}) \end{aligned} \tag{5.1}$$

where the probability density function $p_\psi(.)$ is parameterized by a set of parameters, termed as ψ.

The objective of the CVAE is to maximize the conditional log-likelihood $\log p_\psi(\boldsymbol{x}|\boldsymbol{c})$, i.e., to find the set of parameters ψ that maximizes the log-likelihood of the vector $\boldsymbol{x}$, given the condition variable $\boldsymbol{c}$. This can be implemented by means of a Stochastic Gradient Variational Bayesian (SGVB) framework [94]. Since maximizing the conditional log-likelihood directly is generally intractable, the variational lower bound of the conditional log-likelihood can be used as a surrogate objective function to achieve a more feasible solution [95], which can be written as:

$$\log p_\psi(\boldsymbol{x}|\boldsymbol{c}) \geq -\mathrm{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c}) \parallel p_\psi(\boldsymbol{z}|\boldsymbol{c})\right) + \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})}\left[\log p_\psi(\boldsymbol{x}|\boldsymbol{c},\boldsymbol{z})\right] \tag{5.2}$$

where the operator $\mathrm{KL}\left(q_\phi(.) \parallel p_\psi(.)\right)$ represents the Kullback–Leibler (KL) divergence [36] between the distributions $q_\phi(.)$ and $p_\psi(.)$, and the operator $\mathrm{E}_{q_\phi(.)}[\log p_\psi(.)]$ stands for the expectation of $\log p_\psi(.)$ based on the distribution $q_\phi(.)$. The distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})$, parameterized by ϕ, is set to approximate the true posterior distribution $p_\psi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})$. The first term on the right-hand-side of Eq. (5.2), i.e., the KL divergence between the approximated posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})$ and the conditional prior $p_\psi(\boldsymbol{z}|\boldsymbol{c})$, provides an indicator of how close the posterior distribution is to the prior distribution; the maximization for the right-hand-side of Eq. (5.2) forces the 2 distributions to be as close as possible, and so this term functions as a regularization term. The conditional prior is generally set as a standard normal distribution, i.e., $p_\psi(\boldsymbol{z}|\boldsymbol{c}) = N(0, \mathbf{I})$ [94], so as to provide an analytical solution (marginalization) for the KL divergence term. Note

that, although the prior distribution of the latent variable $\boldsymbol{z}$ is constrained by the condition $\boldsymbol{c}$, it is reasonable to relax such a constraint so that the prior of $\boldsymbol{z}$ can be modeled to be statistically independent of the condition $\boldsymbol{c}$, i.e., $p_\psi(\boldsymbol{z}|\boldsymbol{c}) = p_\psi(\boldsymbol{z})$ [90]. The second term on the right-hand-side of Eq. (5.2) can be interpreted as the reconstruction of the input $\boldsymbol{x}$ through the log-likelihood $\log p_\psi(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{z})$ based on the sampled $\boldsymbol{z}$ from the approximated (learned) posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})$.

With the above probabilistic framework, an architecture of the CVAE can be set up, where the estimation of the approximated posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})$ is considered as the encoder of the CVAE, while determining the likelihood $p_\psi(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{z})$ as its decoder. Generally, the Multi-Layer Perceptions (MLPs) [78] can be employed to form the structure of the CVAE, which then consists of 1) an encoder network, 2) a decoder network, and 3) a hidden layer to generate the latent variable $\boldsymbol{z}$ by sampling from the approximated posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})$, as shown in **Figure 5.1 (b)**.

Training the CVAE is aimed to maximize the right-hand side of Eq. (5.2), i.e., to minimize the KL divergence between the approximated posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})$ and the conditional prior $p_\psi(\boldsymbol{z}|\boldsymbol{c})$, while maximizing the reconstruction log-likelihood $\log p_\psi(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{z})$ based on the approximated posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})$. As common practice in optimization problems, maximizing the right-hand side of Eq. (5.2) can be converted into a minimization problem by changing the sign of the entire expression. The objective loss function of the CVAE can be now expressed as:

$$L_{CVAE}(\boldsymbol{x}, \boldsymbol{c}; \psi, \phi) = \mathrm{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c}) \parallel p_\psi(\boldsymbol{z}|\boldsymbol{c})\right) - \mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})}\left[\log\left(p_\psi(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\right)\right] \tag{5.3}$$

Since the goal is to reconstruct the vectors of cepstral coefficients, i.e., a typical regression problem, the Mean Squared Error (MSE) between the input $\boldsymbol{x}$ and the reconstructed output $\hat{\boldsymbol{x}}$ can be used to substitute the reconstruction term $\mathrm{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})}\left[\log\left(p_\psi(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\right)\right]$.

In summary, the objective of the CVAE is to jointly optimize the encoder parameter set ϕ, to achieve a distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})$ as close as possible to $p_\psi(\boldsymbol{z}|\boldsymbol{c})$, and the decoder parameter set ψ, to reduce the reconstruction loss. Hence, the final objective loss function can be then expressed as:

$$L_{CVAE}(\boldsymbol{x},\boldsymbol{c};\psi,\phi) = \mathrm{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c}) \parallel p_\psi(\boldsymbol{z}|\boldsymbol{c})\right) + \mathrm{MSE}_\psi(\boldsymbol{x},\hat{\boldsymbol{x}}) \tag{5.4}$$

The implementation details of modeling the CVAE and the data augmentation process will be discussed in Section 2.5.

5.2.3.3 A new strategy for defining the condition variable $\boldsymbol{c}$

The condition variable $\boldsymbol{c}$ can be interpreted as a representation of a set of specific data samples, e.g., a class label of the data obtained from a specific structural condition, which is generally modeled as a categorical variable following a multinomial distribution, expressed as:

$$\boldsymbol{c} \sim M_k(n; p_1, p_2, \ldots, p_k) \tag{5.5}$$

where the discrete probabilities p_i (for $i = 1, \ldots, k$) represent the probabilities of the occurrence for each of the k classes, with n denoting the total number of events. In the study that proposed the CVAE, this condition variable $\boldsymbol{c}$ was assigned by simple categorical values (i.e., 0, 1, 2, …) to represent various class labels of data samples. However, in our case, when faced with data from real-life structural systems, we do not know all the class labels of given data samples in advance, because of some unknown structural conditions. Besides, it is sometimes difficult to clearly define the boundaries among various damage scenarios (i.e., progressive damage conditions) to explicitly label the collected data, leading to a mixture of different damage scenarios.

In this work, an unsupervised learning strategy to define the condition variable vector $\boldsymbol{c}$ is proposed, where $\boldsymbol{c}$ is set equal to the mean of the samples of a dataset to be augmented. This mean can be representative of a specific damage scenario, or of a mixture of several consecutive damage

scenarios linked to the dataset. Let us consider a structure that is monitored at N_d locations and that N events have been monitored. Consequently, the available dataset can be represented as $\{\boldsymbol{x}_{i,d}\}_{i=1;\,d=1}^{N;\,N_d}$. Such a dataset accounts for $N \cdot N_d$ samples $\boldsymbol{x}_{i,d} \in R^Q$ ($i = 1, \dots, N$ and $d = 1, \dots, N_d$), where each sample $\boldsymbol{x}_{i,d}$ contains the Q original cepstral coefficients $c_{i,d}[q]$ for $q = 1, \dots, Q$ extracted from the i^{th} record of the acceleration response at the d^{th} recording location. Then, the condition variable $\boldsymbol{c}$ is dependent on the location d and so it is denoted as $\boldsymbol{c}_d$ ($d = 1, \dots, N_d$); it can be interpreted as the location-dependent mean of the vectors $\boldsymbol{x}_{i,d}$ for $i = 1, .., N$, defined as:

$$\boldsymbol{c} = \boldsymbol{c}_d = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{i,d} \tag{5.6}$$

In such a way, the condition variable vector $\boldsymbol{c}_d$ can help preserve local characteristics of the cepstral coefficients at the location d when generating new data samples.

When using the weighted cepstral coefficients in $\boldsymbol{x}_i'$, the condition variable vector $\boldsymbol{c}$ then becomes independent to the location d, which is thus termed as the simple mean of the vectors $\boldsymbol{x}_i'$ for $i = 1, .., N$ (i.e., the global mean of all data samples in the training set), defined as:

$$\boldsymbol{c} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i' \tag{5.7}$$

Accordingly, when generating new data samples for the weighted cepstral coefficients, there will be no consideration of the local characteristics of the cepstral coefficients from a particular location.

5.2.4 Probabilistic linear discriminant analysis

In this work, the Probabilistic Linear Discriminant Analysis (PLDA) [91] model, built by the training dataset augmented by the CVAE, is employed to perform structural damage identification and classification. The PLDA is a probabilistic version of the Linear Discriminant Analysis (LDA) [96] that is a classical technique for data dimension reduction and classification in a supervised-learning strategy (to be introduced in Section 6.2.2). Readers can refer to [91] for details about the derivation of the PLDA.

An important distinction between the PLDA and the LDA is that the former can be used to handle classification tasks for data classes that are not present in the training dataset. This is extremely important in SHM analysis of civil structures because available databases are usually comprised of data from undamaged or from unknown structural conditions. For convenience, in the following illustration, classes of data that appear in the training dataset of the PLDA are referred to as "seen" classes, while those that are not contained in the training dataset are referred to as "unseen" classes.

For parameter optimization of the PLDA model, the study in [91] has provided a closed-form mathematical derivation that can analytically solve the optimization problem based on a maximum-likelihood framework, with the prerequisite that each of the classes in the training dataset contains the same number of samples. As previously noted, in real-life damage classification problems, this is generally not the case due to limited data available for different structural damage conditions. Hence, the data augmentation strategy proposed in this chapter can help solve the problem of the unbalanced training dataset by generating additional samples for the limited data of various structural damage conditions, so that there is an equal number of training samples in each class.

The classification of "seen" classes is a typical supervised-learning problem that can be solved by determining which class the testing data should belong to. This can be done by calculating the likelihood that the testing data belong to each class separately and then assigning the data to the class with the maximum likelihood. In contrast, the classification of the "unseen" classes becomes an unsupervised-learning problem and can be solved through hypothesis testing [97]. Specifically, two likelihoods, corresponding to the probability that a testing dataset belongs and does not belong to a "seen" class in the training dataset, are first determined. Then the logarithm of the ratio (termed as log-likelihood ratio $\ln R$) between these two likelihoods is calculated. Under the assumption that the prior probabilities of two datasets belonging and not belonging to the same class are equal, a positive value of the log-likelihood ratio $\ln R$ indicates the two datasets belong to the same class. On the contrary, a negative value of $\ln R$ represents that the testing data cannot fit within the class considered. If the testing data do not match with any previous classes, then the new testing data will be corresponding to a different structural condition, never "seen" before.

5.2.5 Implementation of data augmentation-based damage classification strategy

The implementation for the proposed data augmentation-based damage classification strategy consists of two main components: First, a CVAE architecture is built and trained for augmenting the original unbalanced training dataset so to obtain a well-balanced training dataset that not only has enough training samples for all the damage scenarios, but also contains an equal number of samples in each of the classes. Second, the augmented training dataset is subsequently used to better train a PLDA model whose parameters can be analytically optimized. In this work, a sliding-window strategy to timely update the PLDA model is proposed so to handle the practical case

where the training dataset is continuously updated over time because of new data recorded by the monitoring system.

5.2.5.1 CVAE hyperparameters

As introduced in Section 2.3, the Multi-Layer Perceptrons (MLPs) were employed to build the CVAE architecture, with the hyperparameters shown in **Table 5.1**. These values were selected based on a series of trial-and-error calibration and on the rules of thumb in [76]. It is noteworthy that the cepstral coefficients, by virtue of their compact representation of the structural properties, greatly simplified the structure of the built CVAE architecture, thus speeding up the training and data generation processes with much less computationally demanding efforts compared to existing deep-learning methods used in vision-based structural health monitoring frameworks [88]. The requirements for computational resources are described in Section 5.2.6.

Table 5.1: The calibrated hyperparameters used for building the CVAE architecture.

Property	Value
Input/Output layer size	50
Intermediate layer size	32
Hidden layer size	10
Activation function (intermediate/hidden layer)	Sigmoid
Activation function (Output layer)	Identity
Epoch	200
Batch size	32
Learning rate	1e-3
Optimizer	Adam

5.2.5.2 CVAE training and data augmentation

To introduce the implementation details of the training process of the CVAE and of the proposed data augmentation strategy, let us first consider a situation where an initial training

dataset $\left\{\boldsymbol{x}_{i,k}^{tr}\right\}_{i=1;\,k=1}^{N_k^{tr};\,K}$ is obtained from a structural system that has already experienced K known damage scenarios (i.e., K "seen" damage classes). This initial training dataset consists of N_k^{tr} samples $\boldsymbol{x}_{i,k}^{tr} \in R^Q$ $(i = 1, \dots, N_k^{tr})$ for each damage class k $(k = 1, \dots, K)$, where each vector sample $\boldsymbol{x}_{i,k}^{tr}$ contains the Q considered cepstral coefficients (either the original ones or the weighted ones). The value of the condition variable $\boldsymbol{c}_k^{tr}$ corresponding to the class k in the training dataset can be set by using one of the 2 strategies presented earlier, i.e., 1) using the categorical values or 2) using the mean vector of the samples in class k (Section 5.2.3.3). Our goal is to augment this initial training dataset with additional simulated sample vectors so that each damage class has the same number of samples, i.e., N_k^{tr} $(k = 1, \dots, K) = N^{tr}$. This allowz us to better train the PLDA model by the analytical solution for optimizing the model parameters (Section 5.2.4).

The CVAE-based data augmentation can be performed either for all the K classes simultaneously or for each individual class separately. The former approach is adopted in this work. To generate new training data samples for all the K classes, the initial training dataset $\left\{\boldsymbol{x}_{i,k}^{tr}\right\}_{i=1;\,k=1}^{N_k^{tr};\,K}$ is first used to train the CVAE architecture with the hyperparameters shown in **Table 5.1**. Based on this initial training dataset and on the theory discussed in Section 5.2.3.2, the objective loss function of the CVAE can be expressed as:

$$\begin{aligned} & L_{CVAE}\left(\left\{\boldsymbol{x}_{i,k}^{tr}\right\}_{i=1,\;\;k=1}^{N_k^{tr},K}, \{\boldsymbol{c}_k^{tr}\}_{k=1}^{K}; \psi, \phi\right) \\ & = \sum_{k=1}^{K}\sum_{i=1}^{N_k^{tr}}\left\{\mathrm{MSE}_{\psi}\left(\boldsymbol{x}_{i,k}^{tr}, \hat{\boldsymbol{x}}_{i,k}^{tr}\right) + \sum_{h=1}^{H} KL\left(q_{\phi}\left(z_{i,k}^{tr}[h]\middle|\boldsymbol{x}_{i,k}^{tr}, \boldsymbol{c}_k^{tr}\right) \parallel N(0,1)\right)\right\} \end{aligned} \tag{5.8}$$

where H represents the hidden layer dimension, which is selected equal to 10 in this work. $z_{i,k}^{tr}[h]$ represents the h^{th} element of the sample $\boldsymbol{z}_{i,k}^{tr}$ that is sampled from the approximated posterior distribution $q_{\phi}(.)$ by using the reparameterization trick introduced in [94],

corresponding to the i^{th} data sample of the k^{th} class. The KL loss (the second term in the curly brackets on the right-hand-side of Eq.(5.8)) in this case is equivalent to the sum of all KL divergences between each one-dimensional component $q_\phi\left(z_{i,k}^{tr}[h]\middle|\boldsymbol{x}_{i,k}^{tr}, \boldsymbol{c}_k^{tr}\right)$ $(h = 1, .., H)$ of the approximated posterior distribution $q_\phi(.)$ and the standard normal distribution $N(0, 1)$.

After completing the training process, the decoder of the CVAE is separated and used to generate $N^{tr} - N_k^{tr}$ new samples for each class k $(k = 1, \dots, K)$. For each class k, $N^{tr} - N_k^{tr}$ new samples of the latent variable $\boldsymbol{z}_{i,k}^{new}$ $\left(i = 1, \dots, (N^{tr} - N_k^{tr})\right)$ are sampled from the approximated posterior distribution $q_\phi(.)$. Then, the newly generated $N^{tr} - N_k^{tr}$ samples $\boldsymbol{z}_{i,k}^{new}$ and the condition term $\boldsymbol{c}_k^{tr}$ are input to the decoder to generate $N^{tr} - N_k^{tr}$ new samples $\boldsymbol{x}_{i,k}^{new}$ $(i = 1, \dots, (N^{tr} - N_k^{tr}))$. After finishing the data generation for all the K classes, the initial training dataset $\left\{\boldsymbol{x}_{i,k}^{tr}\right\}_{i=1;\, k=1}^{N_k^{tr};\, K}$ and the newly generated dataset $\left\{\boldsymbol{x}_{i,k}^{new}\right\}_{i=1;\, k=1}^{N^{tr}-N_k^{tr};\, K}$ are combined into a new augmented dataset $\left\{\boldsymbol{x}_{i,k}^{tr}\right\}_{i=1;\, k=1}^{N^{tr};\, K}$ that will be used to train the PLDA model (Section 5.2.4).

5.2.5.3 A sliding-window strategy for damage classification

Let us now consider the practical implementation of the algorithm in real-life problems where a structural system is continuously monitored, i.e., data that may or may not represent new damage scenarios are continuously acquired. Hence, we propose a sliding-window strategy for damage identification and classification, where the CVAE and PLDA models are constantly updated with the latest "new" data to timely expand their knowledge of the latest structural conditions.

The previous work in Chapter 4 have presented the novel unsupervised-learning method, based on the autoencoders with proper statistical-pattern-recognition strategies, for a binary structural damage classification (i.e., a damage-or-not classification). In this method, a training

distribution of the DSFs representing the undamaged state of a monitored structural system is first established and subsequently used to determine whether newly acquired testing samples are obtained from the system in a damaged state or not. In this work, we focus on the case of multi-class damage classification problems, where the number of the known types (classes) of structural conditions is greater than one, i.e., including the undamaged condition and at least one class of damaged scenarios. The proposed sliding-window strategy can be described as follows:

1) Consider an initial training dataset, with N_1^{tr} ($k = 1$) training samples obtained from the system in undamaged condition, denoted as $\{\boldsymbol{x}_{i,1}^{tr}\}_{i=1}^{N_1^{tr}}$, and N_2^{tr} ($k = 2$) training samples from one known structural damage scenario, denoted as $\{\boldsymbol{x}_{i,2}^{tr}\}_{i=1}^{N_2^{tr}}$. Let us assume the realistic case where there is a sufficient number of samples from the undamaged condition while the set of samples corresponding to the damage condition is small and need to be augmented (i.e., $N_2^{tr} < N_1^{tr} = N^{tr}$). Using the proposed CVAE model, $N^{tr} - N_2^{tr}$ new samples for the damaged scenario can be generated so that $\{\boldsymbol{x}_{i,2}^{tr}\}_{i=1}^{N_2^{tr}}$ becomes $\{\boldsymbol{x}_{i,2}^{tr}\}_{i=1}^{N^{tr}}$. At this point, the two datasets corresponding to the damaged and undamaged scenarios can be considered as representative of the two "seen" classes of data and used to train a PLDA model, with the analytical solution to optimize its parameters.

2) When a new dataset $\{\boldsymbol{x}_{i,j}\}_{i=1}^{N_j}$ is acquired from the system in an unknown scenario, we firstly check whether this new dataset belongs or not to one of the existing classes. This is done by computing the log-likelihood ratios between this new set and each of the previous training sets (Section 2.4). If the newly obtained data belong to one of the 2 existing classes, the corresponding class label will be assigned to them (i.e., $j = 1$ or 2) and we move directly to the next stage to acquire the next round of new data from the system. If the current dataset does not fit in any of the previous ones (i.e., an "unseen" class), then a new label $j = 3$ will be assigned to it, followed by

splitting the dataset into two subsets, namely, one $\{\boldsymbol{x}_{i,j}^{te}\}_{i=1}^{N_j^{te}}$ for testing, and the other $\{\boldsymbol{x}_{i,j}^{tr}\}_{i=1}^{N_j^{tr}}$ for generating new training samples, with $N_j^{te} + N_j^{tr} = N_j$. The set $\{\boldsymbol{x}_{i,3}^{tr}\}_{i=1}^{N_3^{tr}}$ will be used to retrain the CVAE to generate new samples so that $\{\boldsymbol{x}_{i,3}^{tr}\}_{i=1}^{N_3^{tr}}$ becomes $\{\boldsymbol{x}_{i,3}^{tr}\}_{i=1}^{N^{tr}}$. The remaining subset $\{\boldsymbol{x}_{i,3}^{te}\}_{i=1}^{N_3^{te}}$ will be re-tested over the augmented set $\{\boldsymbol{x}_{i,3}^{tr}\}_{i=1}^{N^{tr}}$ to validate the accuracy of the generated samples. Afterwards, the PLDA model will be re-trained over the updated training data.

A flowchart summarizing this sliding-window strategy is given by **Figure 5.2**. Its effectiveness has been validated by the experimental data of a real bridge structure, the results of which are presented in Section 5.3.2.

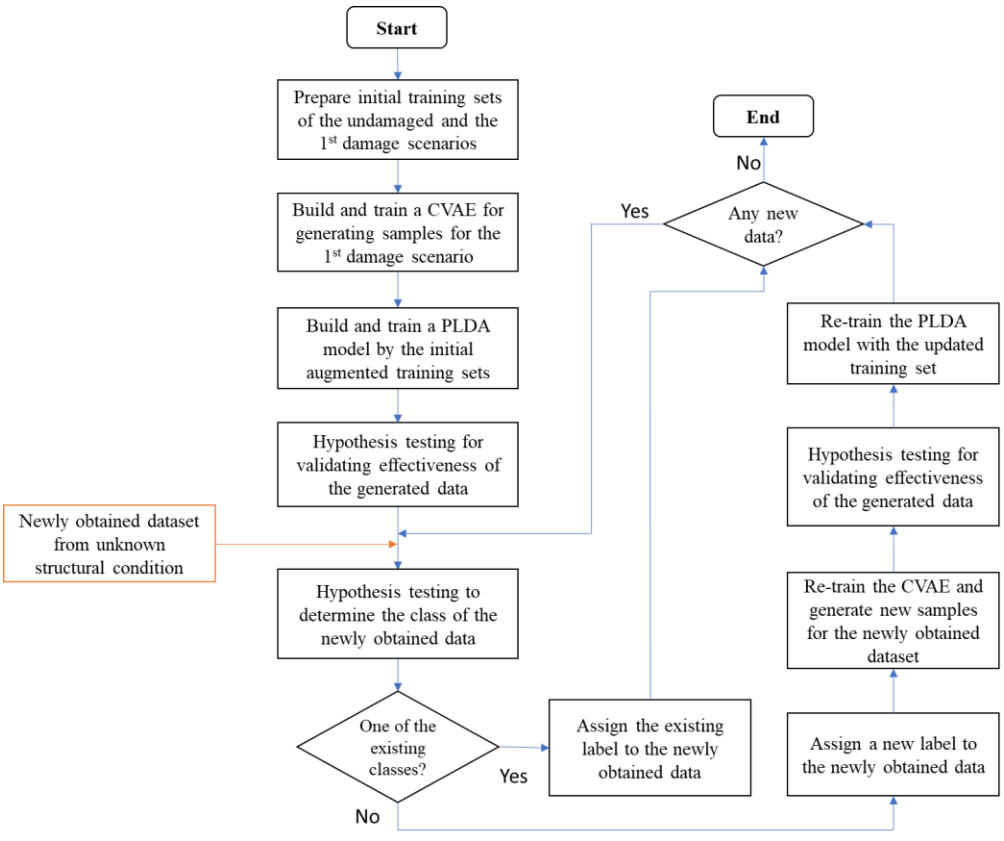

Figure 5.2: A flowchart of the proposed sliding-window strategy for structural damage identification and classification.

5.2.6 Computational requirements

The built CVAE architecture, for generating data samples of the cepstral coefficients, can be used in rapid data augmentation tasks (Section 5.3) with moderate computational requirements because of its concise and easy-setup structure. The built PLDA model, as a non-deep learning probabilistic classifier, also does not require excessive computational resources. The CVAE architecture and the PLDA model evaluated were run on a standard computer with Intel (R) core (TM) 3.89 GHz CPU and 16 Gb of memory. The code was written in MATLAB (for the cepstral-coefficient extraction) and Python 3 (for the CVAE and PLDA modeling). The CPU time for the entire process of the coefficient extraction and the training for the CVAE and PLDA model is about 300-350 s for the numerical case study of the 8 DOF system (Section 5.3.1), and 150-200 s for the case study of the Z24 bridge (Section 5.3.2). Therefore, a standard machine with one CPU could easily provide the required computational power and speed needed in real-life damage assessment applications.

5.3. Numerical and Experimental Analyses

Two case studies were conducted to validate the effectiveness of the proposed data augmentation and damage classification methods. In the first case study (Section 5.3.1), cepstral coefficients were extracted from the simulated time histories of the structural acceleration of the 8 DOF shear-type discrete model (**Figure 4.4**) considering a variety of structural conditions. In the second case study (Section 5.3.2), the cepstral coefficients were again obtained from the recorded acceleration response of the Z24 bridge (Section 4.3.2).

5.3.1 8 DOF shear type system – Case 1

The lumped mass model of an 8 DOF shear-type system, shown in **Figure 4.4**, is again analyzed in this work. The baseline stiffness of the vertical elements is set to $k_d^0 = 25{,}000$ N/m ($d = 1, \dots, 8$), and each mass is equal to $m_d = 1\ kg$ ($d = 1, \dots, 8$). The assumption of modal damping is used, assigning a damping factor of $\xi = 1\%$ for each of the 8 vibrational modes. To simulate different operational and damage conditions, the same sixteen different scenarios as shown in **Table 4.3** were considered. For each scenario, the excitation is provided by 8 different external forces applied at the 8 DOFs; these forces are all modeled as zero-mean Gaussian white noise signals with the zero-order-hold (ZOH) assumption. Their magnitudes are set to increase gradually and linearly from the bottom DOF (1st DOF) to the top DOF (8th DOF), with values from 100 N to 800 N. Each realization of the forces has a duration of 500 seconds with a sampling period of 0.005 seconds (200 Hz sampling frequency). The generated acceleration time histories at the 8 DOFs are then corrupted by a 10% RMS Gaussian white noise to simulate measurement error.

A total of 900 realizations of the acceleration response were simulated for the 9 undamaged scenarios (100 ones for each of the 9 scenarios); for each scenario, 80 realizations were randomly selected to form the training set corresponding to the undamaged conditions (a total of 720 realizations), while the remaining 20 realizations were used as testing set (a total of 180 realizations). When using the original cepstral coefficients from every DOF, there were a total of $720 \times 8 = 5760$ sequences of the original cepstral coefficients extracted, which thereby formed an undamaged training set of 5760 sample vectors $\boldsymbol{x}_{i,d}$ ($i = 1, \dots, 720$ and $d = 1, \dots, 8$). Alternatively, using the weighted cepstral coefficients, a total of 720 sample vectors $\boldsymbol{x}_i'$ ($i = 1, \dots, 720$) were created to form the undamaged training set. Similarly, for the undamaged testing

set, a total of $180 \times 8 = 1440$ samples of $\boldsymbol{x}_{i,d}$ were obtained for the original cepstral coefficients, while this number reduces to a total of 180 samples of $\boldsymbol{x}_i'$ for the case of the weighted cepstral coefficients. With respect to each of the 7 damage scenarios, 100 realizations of acceleration responses were simulated to mimic the real-life situation where the unbalanced training data with limited ones in damaged scenarios are typically obtained from the monitored structural system. Half of those were randomly selected as the initial training dataset, while the remaining half ones were used for testing.

In this analysis, the first 4 damage scenarios (i.e., scenarios 10 – 13) were selected as the 4 "seen" damage classes, which were used to train the CVAE and to generate new samples for the 4 damage classes. The training set of the undamaged class (i.e., scenarios 1 – 9) and the 4 augmented training sets of the 4 "seen" damage classes were used to train the PLDA model, which was then tested using the corresponding 5 testing sets. This trained PLDA model was then used to identify the 3 "unseen" damage classes, i.e., scenarios 14 – 16, through the hypothesis testing strategy (Section 5.2.4).

Before investigating the classification performance by the PLDA model, we first checked the accuracy of the new samples generated by the trained CVAE, since this is essential to the implementation of the proposed damage classification process. **Figure 5.3** provides a comparison between the cepstral coefficients in $\boldsymbol{x}_{i,d}$ or $\boldsymbol{x}_i'$, and the corresponding ones generated by the decoder of the CVAE, for the case of scenario 10. For the original cepstral coefficients in $\boldsymbol{x}_{i,d}$ (**Figure 5.3 (a)**), it can be observed that the generated cepstral coefficients differ considerably from the real ones when the condition variable $\boldsymbol{c}$ is defined by the categorical values (0, 1, 2, …), while they are quite consistent with the real ones when $\boldsymbol{c}$ is equal to the location-dependent mean vector of the training set. This is because the condition variable $\boldsymbol{c}$ defined by categorical values

treats all the cepstral coefficients for each location equally, without taking into account that the cepstral coefficients obtained at different recording locations are different. This will end up generating new cepstral coefficients that are close to the global mean of all the data in the training set (Section 5.2.3.3). In contrast, using Eq. (5.6), it will allow us to account for such a variation, thus generating cepstral coefficients that are strongly correlated with the real ones, as shown in the **Figure 5.3 (a)**. When using the weighted cepstral coefficients in $\boldsymbol{x}_i'$ (**Figure 5.3 (b)**), being already a specific weighted average of the coefficients from all DOFs, the real and generated coefficients are quite close to each other, regardless of the strategies used to define the condition variable $\boldsymbol{c}$. Moreover, as discussed in Section 5.2.1.2, the weighted cepstral coefficients are an enhanced representation of the overall structural properties, since a large amount of variance associated with the excitation and measurement noise has been removed (these coefficients follow a more compact Gaussian distribution.). Accordingly, the data augmentation process for the weighted cepstral coefficients is more robust, making it easier to generate accurate new cepstral coefficient datasets.

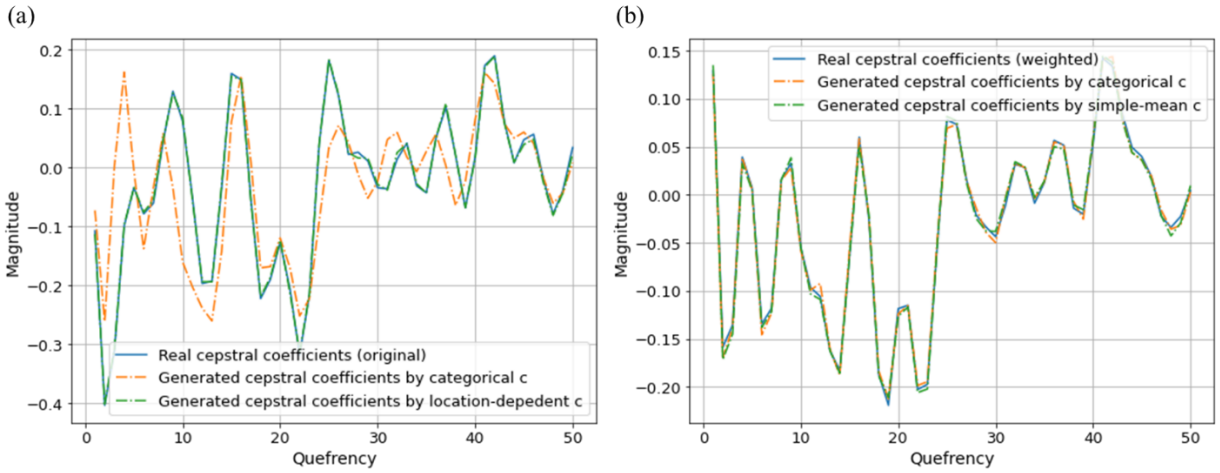

Figure 5.3: A comparison between the real cepstral coefficients and the generated cepstral coefficients. (a) The case with original cepstral coefficients. (b) The case with the weighted cepstral coefficients.

Let us now focus on the results of the damage classification based on the augmented training dataset and on the PLDA model. **Figure 5.4** shows the Receiver Operating Characteristic (ROC)

curves of the damage classification results, produced from the testing sets of the 5 “seen” classes by implementing the 4 introduced strategies, i.e., using the original cepstral coefficients integrated with data augmentation based on the condition variable $\boldsymbol{c}$ as the categorical value (1) or as the location-dependent mean vector (2), and using the weighted cepstral coefficients integrated with augmented data from the condition $\boldsymbol{c}$ as the categorical value (3) or as the simple mean vector (4). The results obtained without the implementation of the data augmentation are presented as well for comparison. It can be observed that the data augmentation strategy improves the classification performance for both types of cepstral coefficients (the original cepstral coefficients and the weighted ones), producing larger Area Under the Curve (AUC) values. All the 4 strategies can achieve excellent classification performances, as all the curves increase rapidly with the false positive rate (the AUC can even reach the perfect score of 1 with the weighted coefficients.). For better validation, we further investigated the results of two classical evaluation metrices, i.e., accuracy and F1-score, over the testing sets of the 5 “seen” classes, that yielded classification results consistent with the ROC curves, as shown in **Table 5.2**. One can observe that the data augmentation strategy integrated with the weighted cepstral coefficients can help achieve perfect performance of 100% accuracy when based on the categorical condition, and almost perfect accuracy when based on the mean-vector condition. It is demonstrated again that the data augmentation is clearly effective and that these weighted cepstral coefficients offer a better representation of the overall structural properties.

Another important observation is that the data augmentation implemented through the categorical condition always produces slightly better classification results than those produced by using the mean-vector condition, regardless of the type of the cepstral coefficients. This is because the PLDA model is trained to recognize each damage scenario categorically, and such a training

process is close to the CVAE-based data augmentation process with the categorical condition. Nevertheless, the generated cepstral coefficients through the categorical condition are not sufficiently correlated to the real ones, as shown in **Figure 5.3** as compared to those generated with the location-dependent condition. Moreover, the labels of the various damage scenarios of a structural system are usually not known explicitly in advance, making almost impossible to use the categorical condition in real-life SHM problems.

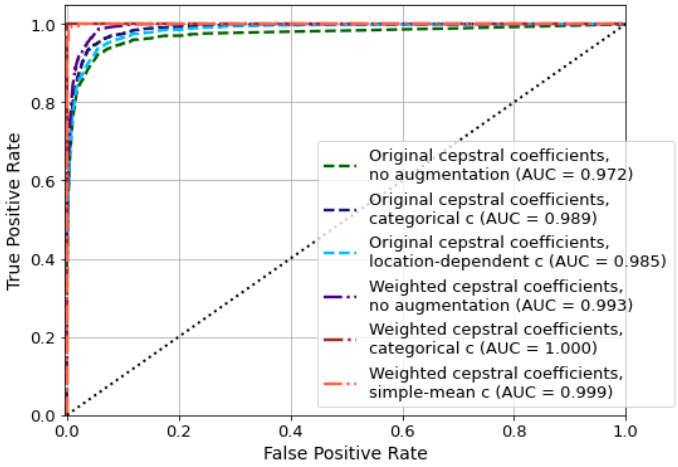

Figure 5.4: Damage classification results of the ROC curves for the 8 DOF system.

After completing the testing for the 5 "seen" classes, i.e., scenarios 1 – 9 (the undamaged class) and scenarios 10 – 13 (the first 4 damage classes), the PLDA model was then tested over the data of the 3 "unseen" damage scenarios, i.e., scenarios 14 – 16, by implementing the hypothesis testing described in Section 5.2.4. **Table 5.3** provides the results of the computed log-likelihood ratios between the testing sets of the 3 "unseen" scenarios and the training sets of all the 16 scenarios, based on the weighted cepstral coefficients. Note that, the training sets of these 3 "unseen" scenarios were not used in the training of the PLDA model. The percentage values in parenthesis represent the percentage of samples of the target testing set that were identified as the same class linked to the corresponding column. The results show that negative log-likelihood ratios

(e.g., not-matching classes) were obtained consistently when implementing the hypothesis testing between each of the 3 "unseen" classes and each of the 5 "seen" classes, with almost perfect performance except for the 2% test samples of scenario 14 being incorrectly identified as of the same class as scenario 13. However, these two scenarios are almost identical, having the same damage location (i.e., the stiffness reduction at DOF 7) but different damage severity (15% vs. 10%). Additionally, all the hypothesis testing between any two of the 3 "unseen" damage classes return negative log-likelihood ratios, indicating that the algorithm can recognize that these 3 different damage scenarios belong to 3 different damage classes. The effectiveness of the generated samples for the 3 "unseen" damage classes is verified by the positive log-likelihood ratios generated by the hypothesis testing between the augmented training sets and the corresponding testing sets (the diagonal elements of the last three columns in **Table 5.3**). These results show that the proposed data augmentation strategy can help the PLDA model achieve excellent performance in the damage identification and classification for the 8 DOF system.

Table 5.2: Damage classification results of the accuracy and F1-score for the 8 DOF system.

	Accuracy (%)	F1-score (%)
Original CCs, no data augmentation	91.357	90.422
Original CCs, categorical values as condition	94.605	93.561
Original CCs, location-dependent mean vector as condition	94.049	93.252
Weighted CCs, no data augmentation	96.316	95.863
Weighted CCs, categorical values as condition	**100.000**	**100.000**
Weighted CCs, simple mean vector as condition	**99.737**	**99.543**

Table 5.3: Log-likelihood ratios between the 3 "unseen" scenarios and all the 16 scenarios.

	Scenarios 1-9 (training)	Scenario 10 (training)	Scenario 11 (training)	Scenario 12 (training)	Scenario 13 (training)	Scenario 14 (training)	Scenario 15 (training)	Scenario 16 (training)
Scenario 14 (testing)	-32154.38 (0.00%)	-32719.66 (0.00%)	-24460.82 (0.00%)	-32839.87 (0.00%)	-7212.98 (2.00%)	7.12 (88.00%)	-5949.35 (4.00%)	-44892.84 (0.00%)
Scenario 15 (testing)	-29382.70 (0.00%)	-34008.35 (0.00%)	-19348.15 (0.00%)	-34844.83 (0.00%)	-17604.05 (0.00%)	-6181.24 (2.00%)	6.44 (86.00%)	-49924.76 (0.00%)
Scenario 16 (testing)	-29419.22 (0.00%)	-36719.30 (0.00%)	-37869.37 (0.00%)	-32532.97 (0.00%)	-42899.40 (0.00%)	-45514.27 (0.00%)	-48953.39 (0.00%)	8.62 (90.00%)

5.3.2 Z24 bridge – Case 2

The data recorded during operation and demolition of the Z24 bridge (Switzerland) were again used in this work to evaluate the performance of the proposed data augmentation and damage classification strategies in dealing with data from real applications. An overview of the various monitoring campaigns, the damage conditions, and the accelerometers placement of the bridge have been introduced in Section 4.3.2. In this analysis, the recorded acceleration responses of the first two scenarios in bold on **Table 5.4** (July 10th – 17th and August 4th – 9th), representative of the undamaged conditions, were labelled as corresponding to the undamaged class: Although the environmental conditions (e.g., temperature and humidity) were quite similar, the data recorded between July 10th – 17th were obtained from the bridge in its regular operational conditions while those from August 4th – 9th correspond to forced vibration tests. The data of the remaining 9 scenarios in bold between August 25 and September 9 were labelled as the ones representing damaged conditions. In the following descriptions, we will refer to these datasets as the data of damage scenarios 1 – 9 in chronological order, rather than in terms of damage severity. This is because the intent is to show here how the proposed sliding-window strategy (Section 5.2.5.3) can

be employed to handle real-life situations where the data acquired from a structure are continuously processed and there is no knowledge in advance of all possible damage classes.

In order to increase the dataset size with more samples, each record was framed into three 5 minutes segments, with 50% overlap. Accordingly, a total of 864 framed records of the undamaged class, for each of the 6 sensors, were available. The training set of the undamaged class was created by randomly selecting 80% of the records of the undamaged class, i.e., a total of 691 records, while the remaining 173 records were used as the undamaged testing set. The undamaged training set was thereby composed by $691 \times 6 = 4146$ samples of the vector $\boldsymbol{x}_{i,d}$ when using the original cepstral coefficients, or 691 samples of the vector $\boldsymbol{x}_i'$ for the weighted ones, while the undamaged testing sets were composed by 1038 samples $\boldsymbol{x}_{i,d}$, or 173 samples $\boldsymbol{x}_i'$. For the data corresponding to damage scenarios 1 – 9, the initial training set and the testing set for each scenario were created by evenly splitting the corresponding framed records. As a summary, **Table 5.5** records the numbers of samples in the initial training sets and testing sets for all the considered scenarios, with respect to the two cases of using the original cepstral coefficients in $\boldsymbol{x}_{i,d}$ and the weighted ones in $\boldsymbol{x}_i'$.

Table 5.4: An overview of the considered structural conditions (the ones in bold) in this analysis.

Date (1998)	Scenario
10-17 July	**Undamaged condition**
4 August	**Undamaged condition**
9 August	Installation of pier settlement system
10 August	Lowering of pier, 20 mm
12 August	Lowering of pier, 40 mm
17 August	Lowering of pier, 80 mm
18 August	Lowering of pier, 95 mm
19 August	Lifting of pier, tilt of foundation
20 August	New reference condition
25 August	**Spalling of concrete at soffit, 12 m^2**
26 August	**Spalling of concrete at soffit, 24 m^2**
27 August	**Landslide of 1 m at abutment**
31 August	**Failure of concrete hinge**
2 September	**Failure of 2 anchor heads**
3 September	**Failure of 4 anchor heads**
7 September	**Rupture of 2 out of 16 tendons**
8 September	**Rupture of 4 out of 16 tendons**
9 September	**Rupture of 6 out of 16 tendons**

Table 5.5: The numbers of samples in the initial training sets and testing sets.

	Original cepstral coefficients		Weighted cepstral coefficients	
	nitial training sets	Testing sets	nitial training sets	Testing sets
Undamaged scenarios	4146	1038	691	173
Damage scenario 1	216	216	36	36
Damage scenario 2	216	216	36	36
Damage scenario 3	864	864	144	144
Damage scenario 4	432	432	72	72
Damage scenario 5	216	216	36	36
Damage scenario 6	726	726	121	121
Damage scenario 7	126	126	21	21
Damage scenario 8	108	108	18	18
Damage scenario 9	162	162	27	27

As noted above, the sliding-window strategy was implemented to continuously train the CVAE and the PLDA models so that they are capable of tracking the varying structural conditions of the bridge. As discussed in Section 5.2.5.3, this work is focused on the multi-class damage

classification problem and the sliding-window strategy was performed as shown in **Figure 5.2**, starting from the point where the training sets of the undamaged scenarios and damage scenario 1 have been acquired and their class labels have already been determined. **Table 5.6** records the hypothesis testing results represented by the log-likelihood ratios obtained by the sliding-window strategy using the weighted cepstral coefficients with the condition $\boldsymbol{c}$ defined by the simple-mean vector. The values in parenthesis represent the percentage of samples in the testing set of each damage scenario identified as belonging to the class of the scenario in the corresponding column. Looking at these results, the following observations can be made: 1) For the hypothesis testing between each of the 9 damage scenarios and the undamaged class, all the returned log-likelihood ratios are negative with only minor percentages of samples (ranging from 1.4% to 2.8%) misclassified as the undamaged class for damage scenarios 1 – 4, confirming that the sliding-window strategy can accurately identify the presence of damage. It is also important to note that the first 2 damage scenarios, representing different amount of concrete spalling, have minor effects on the dynamic characteristics of the overall structures, and so it is expected that the mislabeling for that 2 cases (2.8%) is higher than those corresponding to the other cases. 2) In terms of the hypothesis testing as a tool to verify the accuracy of the new samples generated for each damage scenario, all the returned log-likelihood ratios along the diagonal are positive, indicating that the data augmentation strategy is effective in generating training samples for the various damage scenarios of the Z24 bridge, setting the stage for a better training of the PLDA model. 3) It can be noted that there are 2 positive log-likelihood ratios that correspond to 2 misclassification results. The first occurrence is when testing data from damage scenario 2 are tested against the data from damage scenario 1 (0.77). As previously mentioned, this may be due to the fact that the monitoring campaigns during August 25 – 27 (damage scenarios 1 and 2) are related to low-level damage

conditions of the bridge (spalling of concrete at soffit for the both), leading the PLDA model to identify both as the same class. For the second occurrence (when testing the data of damage scenario 4 against the ones of damage scenario 3), the positive log-likehood ratio (0.96) indicates that the algorithm fails to distinguish between those 2 damage scenarios, even though it correctly recognizes them as damage states. One possible explanation could be that since it is a situation of progressive damage, the damage induced by a 1-meter landslide at the abutment has a much greater impact on the structural characteristics than the failure of the concrete hinge at the support: When the failure of the concrete hinge occurs, its effects could be hidden by those produced by the landslide, and this could confuse the classification process. However, the algorithm is still capable of assigning to these data a different label from the other damage scenarios.

Table 5.6: Log-likelihood ratios produced by the sliding-window strategy over the Z24 bridge.

	Undamaged (training)	DS 1 (training)	DS 2 (training)	DS 3 (training)	DS 4 (training)	DS 5 (training)	DS 6 (training)	DS 7 (training)	DS 8 (training)	DS 9 (training)
DS 1 (testing)	-789.25 (2.8%)	**3.52 (83.3%)**								
DS 2 (testing)	-767.12 (2.8%)	**0.77 (75.8%)**	**2.49 (80.6%)**							
DS 3 (testing)	-811.84 (2.1%)	-64.38 (27.8%)	-45.38 (41.7%)	**3.04 (84.7%)**						
DS 4 (testing)	-825.51 (1.4%)	-272.88 (11.1%)	-117.76 (23.6%)	**0.96 (78.8%)**	**4.61 (87.5%)**					
DS 5 (testing)	-998.35 (0.0%)	-506.25 (5.6%)	-391.71 (8.3%)	-263.81 (22.2%)	-68.46 (33.3%)	**6.13 (88.9%)**				
DS 6 (testing)	-987.69 (0.0%)	-496.68 (6.6%)	-417.16 (8.3%)	-260.02 (23.1%)	-57.05 (37.2%)	-15.67 (47.9%)	**5.96 (89.3%)**			
DS 7 (testing)	-1105.67 (0.0%)	-501.34 (4.8%)	-411.65 (9.5%)	-316.07 (19.0%)	-88.51 (33.3%)	-56.51 (38.1%)	-29.93 (47.6%)	**7.23 (90.5%)**		
DS 8 (testing)	-1208.26 (0.0%)	-464.25 (6.25%)	-395.51 (12.5%)	-324.76 (18.8%)	-49.80 (25.0%)	-57.19 (37.5%)	-18.24 (50.0%)	-13.77 (56.3%)	**8.14 (87.5%)**	
DS 9 (testing)	-1206.48 (0.0%)	-533.57 (3.7%)	375.42 (11.1%)	-280.98 (22.2%)	-37.33 (40.7%)	-50.24 (33.3%)	-41.88 (44.4%)	-17.45 (55.6%)	-5.47 (66.7%)	**10.41 (92.6%)**

After completion of the data augmentation and damage identification for all the 9 damage scenarios through the unsupervised sliding-window strategy, the PLDA model was trained and tested again by the updated datasets to validate the performance of the supervised-learning-based damage classification. Based on the previous results, the datasets were re-arranged as 1 undamaged class and 7 damage classes: The datasets of damage scenarios 1 – 2 were both labelled as damage class 1 while those of damage scenarios 3 – 4 were both labelled as damage classes 2. The other datasets were kept the same. **Figure 5.5** presents the ROC curves (with the corresponding AUC values), for the cases of no data augmentation and data augmentation with the 2 different condition variables, and **Table 5.7** provides the corresponding results of the two evaluation metrices (accuracy and F1-score). It can be observed that the PLDA model, trained by the augmented training datasets, can produce higher accuracy compared to the results obtained using the initial training datasets without data augmentation (the F1-scores are 4.294% – 7.006% higher.). In addition, the use of weighted cepstral coefficients considerably improves the performance of the proposed strategy compared to when the original cepstral coefficients are used (a comparison of 82.173% – 85.018% vs. 90.516% – 91.437% for the F1-score). These results again validate the effectiveness of the proposed data augmentation strategy in improving the classification performance of the PLDA model, and demonstrate the superior qualities of the weighted cepstral coefficients in better identifying statistical distribution patterns among data of different damage scenarios when dealing with real-life structural system, thanks to their enhanced and more stable representation of the overall structural properties.

Table 5.7: Damage classification results of the accuracy and F1-score for the Z24 bridge.

	Accuracy (%)	F1-score (%)
Original CCs, no data augmentation	78.983	78.012
Original CCs, categorical values as condition	86.314	85.018
Original CCs, location-dependent mean vector as condition	83.695	82.173
Weighted CCs, no data augmentation	88.195	87.143
Weighted CCs, categorical values as condition	**92.522**	**91.437**
Weighted CCs, simple mean vector as condition	**91.642**	**90.516**

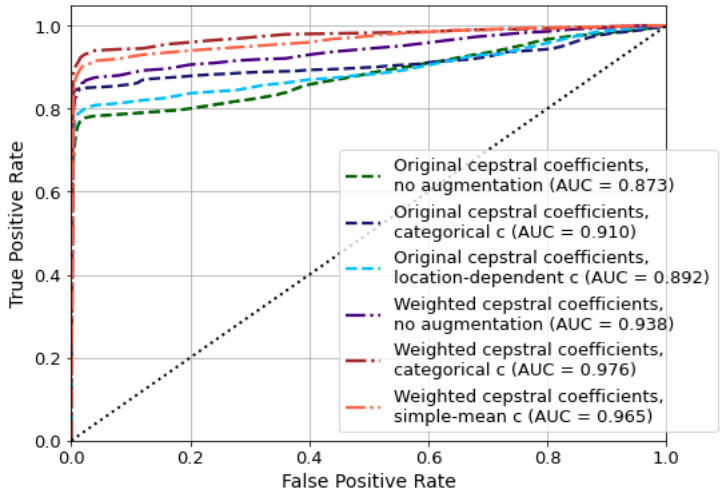

Figure 5.5: Damage classification results of the ROC curves for the Z24 bridge.

5.4. Conclusions

This chapter presents a Structural Health Monitoring (SHM) methodology that is based on a novel data augmentation strategy. Based on a Conditional Variational Autoencoder (CVAE) architecture, this strategy can create a "balanced" dataset of the cepstral coefficients of the structural acceleration response, and use this dataset to systematically build a Probabilistic Linear

Discriminant Analysis (PLDA) model for damage identification and classification. The proposed data augmentation strategy addresses the issue, commonly found in monitoring of real civil structures of limited datasets from structures in damaged conditions. The PLDA model, trained with the augmented balanced dataset, can be performed well for structural damage identification and classification in both supervised- and unsupervised-learning manners. To validate the proposed method, two case studies have been presented: 1) an 8 DOF system model excited by different random Gaussian signals, and 2) a real bridge structure (the Z24 bridge) that was monitored while improving progressive damage. From the analysis of the results, it can be concluded that the proposed data augmentation strategy is able to very effectively augment the training datasets of the cepstral coefficients in various structural damage conditions, and to better train a PLDA model, resulting in an obviously improved performance in damage identification and classification. The following are the main conclusions drawn from this study.

1) Because they provide a compact and effective representation of the structural modal properties, the cepstral coefficients, either in their original definition or in the weighted form, can be efficiently used as features in the proposed data augmentation process, significantly decreasing the complexity of the CVAE architecture. This leads to a significant reduction in overfitting the data and in the required computational resources.

2) For the CVAE modeling, the proposed unsupervised-learning strategy to define the condition random variable of the CVAE, i.e., using a proper mean of a target training dataset for augmentation as condition, can help us handle a common real-life situation where data are obtained from a structure in unknown damage conditions. The weighted cepstral coefficients, because of their enhanced representation of the overall structural properties, supports a more robust training

process for the CVAE that allows a better characterization of the statistical distributions of the cepstral coefficients in the CVAE's encoder latent space.

3) The PLDA-based framework can be implemented in a supervised-learning strategy for the classification of known damage conditions, as well as in an unsupervised strategy for damage conditions not seen before, thanks to a newly developed sliding-window strategy that allows us to identify structural damage conditions not present in the training datasets.

4) For both case studies, the proposed data augmentation strategy is able to effectively augment the initial training dataset of the cepstral coefficients, leading to better performances of damage classification compared to the case of using only the initial training dataset. In addition, the results demonstrate that the data augmentation integrated with the weighted cepstral coefficients can yield excellent damage identification and classification results, with significant improvements over those produced by the original cepstral coefficients.

5) For the Z24 bridge case study, the proposed sliding-window strategy can successfully deal with the common real-life situation where the structural conditions are unknown priori.

Chapter 6. A Discriminant Analysis Strategy for Structural Damage Localization

6.1 Introduction

In Chapters 4 and 5, the theory as well as applicability of the proposed data-driven methods for structural damage detection, quantification, and classification have been presented and extensively discussed. The key intuition of these methods is to model the data, represented by the cepstral coefficients of the structural acceleration response, via deep learning algorithms, i.e., various autoencoder architectures, to better characterize (or simulate) the statistical distribution of the component of the cepstral coefficients that is directly related to the overall structural properties, so as to optimize the damage assessment performance. In this chapter, another topical problem in the field of SHM, i.e., structural damage localization, is systematically studied, and new solutions are proposed for both linear and nonlinear problems in a data-driven perspective.

As introduced in Sections 4.1 and 5.1, advanced data-driven methods developed within the vibration-based SHM framework offer great advantages by providing continuously updated information on the condition of the monitored structural systems. Vibration-based methods (VBMs) for damage quantification and localization have been widely explored in the past decades by analyzing the dynamic response of structures under ambient or forced vibrations [98]. Detecting the occurrence of damage using responses measured by sensors that are not necessarily deployed near the damaged areas (which are unknown a priori) is one of the main advantages of vibration-based damage localization strategies [99]. It is well known that structural damage translates as a loss of stiffness, and the VBMs rely on the fact that a reduction of stiffness leads to changes in the structural response and in the dynamic characteristics (e.g., natural frequencies and modal shapes).

Localized damage can then be described by an appropriate evaluation metric (i.e., a damage index) defined by comparing the response or the structural characteristics in the undamaged state with the ones in the current inspection state of the structure for each of the monitored locations.

Most classical strategies for damage localization fall into the category of the model-based methods. One of the most popular methods is the modal-updating approach using Finite Element (FE) models in which the parameters of the model are continuously updated to minimize an objective function that is a measure of the difference between the recorded structural response from the sensors and the simulated response of the built FE model [100]. Since such analyses can have a significant computational cost as the result of complex iterative optimization process, many scholars have turned to data-driven methods for achieving more efficient solutions while considerably reducing the intrinsic complexity and computational cost of the model-based methods [54, 6]. Data-driven methods are typically developed by directly analyzing and modeling the recorded structural response data. By extracting appropriate Damage Sensitive Features (DSFs) from the response, these methods are able to greatly accelerate the training and inference processes of the model as introduced in Chapters 4 and 5.

Among the data-driven methods, one effective strategy for damage localization problems is to locate damage by looking at the geometric changes in structural modal characteristics induced by the local reduction of stiffness. Using only structural response data, various data-driven identification algorithms have used mode shapes and their curvature information for damage localization problems [101]. The mode shapes and their curvatures have the advantage that they provide essential structural spatial information that can be used, directly and indirectly, for damage localization tasks, and their effectiveness has been proven in applications with beam-like and plate-like structures [102]. The essential idea of these methods is to define an appropriate DSF, related

to the identified mode shape, and then use it to determine a damage indicator/index (e.g., difference or ratio) to reflects local (geometric) variations in the structure due to changes in its structural condition. Some of the most relevant methods include the curvature method [103], the strain method [104] and the interpolation error method [105]; the study in [98] provide a comprehensive comparison on the effectiveness of these methods.

In recent years, scholars have been developing data-driven damage localization methods through state-of-the-art signal processing and Machine Learning (ML) techniques. Wavelet analysis, as a powerful time-variant signal processing technique, has been widely used for damage localization in beam-like structures [106, 107]. Solis et. al [107] employed the continuous wavelet transform on the identified mode shape vectors to obtain information on the curvature change of beams between the reference undamaged state and the potentially damaged state. Besides, as presented in [108], the SHM techniques based on the Lamb waves [109] show great promise for damage localization problems in plate-like structures, such as aircraft wings, wind turbines, and pipeline systems, since Lamb waves can propagate over long distances and are sensitive to heterogeneity near the propagation path. Recently, Zhang et. al [110] developed a one-dimensional Convolutional Neural Network (CNN) architecture to effectively extract high-level features of the Lamb-wave signals from plate-like structures, and further built a mapping from these features to the damage locations via a regression framework. The CNN in [110] is able to effectively correlate temporal information embedded in wave signals with a significant generalization capability, and can be trained with data from a single plate and then applied (transferred) to a new plate with appreciable accuracy.

However, current research in the field of damage localization, from the development of proper theoretical methods to their validations on structures, still suffers from many limitations. First,

because only a small number of sensors are usually deployed on a structure, the datasets are generally small and this has an impact on the training of the algorithms. Second, since many of the monitored structures have never experienced damage, the response datasets are usually comprised only by data obtained from the structure in undamaged conditions and so cannot be used for damage localization. Because of these reasons, most localization methods in the literature have been validated only with simulated data from numerical models or with tests on laboratory scale specimens [98]. In addition, many of the proposed damage localization methods are specific to a particular structure, such as the above-mentioned beam or plate structures, or even just to a particular case study [110]. Furthermore, many of the cutting-edge data-driven methods, especially ML-based damage localization modeling, still cannot separate outside a supervised-learning framework that enables the model to acquire structural damage information in advance, i.e., using the data from various damage scenarios for model training. Such supervised learning-based methods typically result in a damage localization procedure that is quite similar to the damage classification problem discussed in Chapter 5. In practice, what we would like to achieve is to train the model using only data from the undamaged state of the structure and when new testing data from an unknown damage state becomes available, the model can first determine whether damage has occurred or not and subsequently indicate the location of the damaged area.

In this chapter, a novel data-driven damage localization method is proposed, which is based on the Linear Discriminant Analysis (LDA) [96] and uses the cepstral coefficients of the structural acceleration response as DSFs. The essential idea is as follows: First, an LDA-based data-driven model is built to emphasize the local structural characteristics embedded in the cepstral coefficients of the acceleration response recorded at multiple locations on the monitored structure in its undamaged state (i.e., an "undamaged" training dataset). These local characteristics are an

enhanced representation of the structural mode shape-related information within the cepstral coefficients. At this point, when a new (testing) set of the cepstral coefficients becomes available, these cepstral coefficients will be passed through the previously built LDA model so to emphasize the local characteristics of the cepstral coefficients of this new dataset. With the help of a properly defined damage index, these two enhanced representations of the two datasets are then compared with respect to every recording location to localize the area of potential damage within the structure. Since only data from the undamaged condition of the structure need to be used in the LDA modeling process, the proposed damage localization method can be implemented in a fully unsupervised-learning manner without requiring the model to access any structural damage information during the training process.

Furthermore, in this chapter, it is proposed to extend this LDA-based modeling method to the damage localization problem for nonlinear structural systems. First, the vibration behavior of a single-degree-of-freedom (SDOF) nonlinear system is analyzed. Then, a validation of the effectiveness of the proposed LDA-based damage localization method in the case of a multi-directional (MDOF) nonlinear system is performed. The motivation for this study rises from the consideration that damage can occur in a progressive fashion, making the structure a time-varying system exhibiting nonlinear dynamic behavior.

6.2 Methodology

6.2.1 Local information from the cepstral coefficients of acceleration response

As introduced in Section 4.2.1, the power cepstral coefficients derived from the acceleration response can provide an alternative representation of the structural characteristics such as natural frequencies, damping ratios, and mode shapes.

As shown in Eqs. (4.6) - (4.7), the cepstral coefficients of the acceleration response, i.e., $c_d[q]$ for $q = 1, \dots, Q$, can be decomposed into two parts, i.e., $\theta[q]$ and $\gamma_d[q]$, where the index q represents the q^{th} cepstral coefficient in the quefrency domain and the subscript d represents the d^{th} monitoring location ($d = 1, \dots, N_d$). As introduced in Section 4.2.1, $\theta[q]$ is associated with the overall structural properties including the natural frequencies and damping ratios, which are independent of the location d, while $\gamma_d[q]$ also contains local structural characteristics such as the modal components at the sensor and actuator locations. The local characteristics in $\gamma_d[q]$ are given by the roots $Z_l^{(d)}$ for $l = 1, \dots, M$, which are the solution of the polynomial shown in Eq. (4.4). By observing the form of the polynomial, it is clear that the roots $Z_l^{(d)}$ for $l = 1, \dots, M$, with respect the d^{th} location, are related to the mode-shape element term $\phi_{d,l}$, which thus are distinct across the N_d locations. The solution of the polynomial for the root $Z_l^{(d)}$ involves a large series of complex terms of $\phi_{d,l}\phi_{j,l}$, where d and j represent the sensor and actuator location, respectively, while l is a summation index over all the contributing modes [111].

Consequently, implementing a proper strategy to highlight the local characteristics related to the mode shapes within the term $\gamma_d[q]$ would be useful for the damage localization. The objective now turns to the development of an effective data-driven method that can enhance the information of the local characteristics in $\gamma_d[q]$, and use it toward damage localization.

6.2.2 A linear discriminant analysis-based strategy for damage localization

To effectively highlight the mode shape-related (local) characteristics within the cepstral coefficients, the Linear Discriminant Analysis (LDA) is considered. The LDA is a robust technique for data analysis first proposed by R. Fisher, to discriminate different types of flowers [112]. This approach is driven by the idea of determining a lower dimensional latent space, compared to the

dimensionality of the original data samples, in which these data samples associated with various categories can be well separated. **Figure 6.1** provides a straightforward visualization of the mechanism of transforming the 2-dimensional data samples with 2 classes into a 1-dimensional latent space through the LDA, where the separation distance between the 2 classes is maximized.

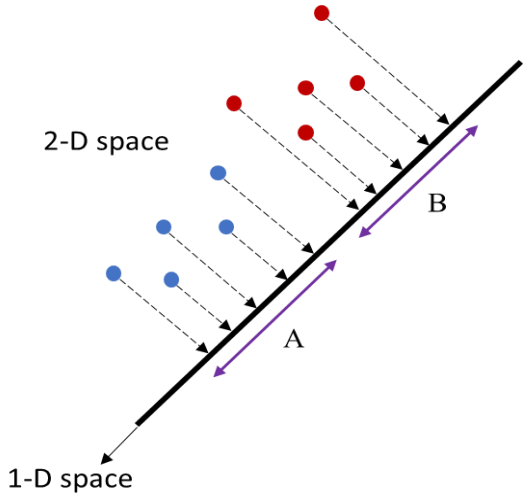

Figure 6.1: A simple example for the intuition behind the LDA. The 2-dimensional data samples are projected in a lower 1-dimensional space, in which the separation between the 2 classes is maximized.

Let us now briefly discuss the mechanism of LDA and show how it can be effectively adapted to the damage localization problem. Consider that a training dataset $\{\boldsymbol{x}_{i,d}^{tr}\}_{i=1;\,d=1}^{N^{tr};\,N_d}$ is obtained from a monitored structural system in its undamaged state, with a total of N_d recording locations (DOFs) and a total of N^{tr} data samples per location (the superscript "tr" represents the case of training data.). The data sample $\boldsymbol{x}_{i,d}^{tr} \in R^Q$ is a vector consisting of Q cepstral coefficients $c_{i,d}[q]$ for $q = 1, \ldots, Q$ with respect to the location d ($d = 1, \ldots N_d$). The data samples of this training set can be then treated as N_d classes by considering each of the recording locations as one class, and each of the N_d classes has an equal number N^{tr} of samples. For convenience, we define C_d^{tr} as the set of the training samples of class d (i.e., the location d), with N^{tr} representing the total number of

samples in the set C_d^{tr}. Subsequently, an LDA model can be built through the training dataset with the goal of maximizing the between-class separation of the training data while minimizing the withing-class scatters. Mathematically, the within-class and between-class scatter matrices with respect to the training data are computed as follows:

$$\boldsymbol{S}_w^{tr} = \frac{\sum_{d=1}^{N_d} \sum_{i=1(i \in C_d^{tr})}^{N^{tr}} \left(\boldsymbol{x}_{i,d}^{tr} - \overline{\boldsymbol{x}}_d^{tr}\right)\left(\boldsymbol{x}_{i,d}^{tr} - \overline{\boldsymbol{x}}_d^{tr}\right)^T}{N^{tr} \cdot N_d} \tag{6.1}$$

$$\boldsymbol{S}_b^{tr} = \frac{\sum_{d=1}^{N_d} N^{tr} (\overline{\boldsymbol{x}}_d^{tr} - \overline{\boldsymbol{x}}^{tr})(\overline{\boldsymbol{x}}_d^{tr} - \overline{\boldsymbol{x}}^{tr})^T}{N^{tr} \cdot N_d} \tag{6.2}$$

where $\overline{\boldsymbol{x}}_d^{tr} = \frac{1}{N^{tr}} \sum_{i=1(i \in C_d^{tr})}^{N^{tr}} \boldsymbol{x}_{i,d}^{tr}$ represents the mean of the training samples of class d, and $\overline{\boldsymbol{x}}^{tr} = \frac{1}{N_d \cdot N^{tr}} \sum_{d=1}^{N_d} \sum_{i=1(i \in C_d^{tr})}^{N^{tr}} \boldsymbol{x}_{i,d}^{tr}$ represents the global mean of all the training samples. $\boldsymbol{S}_w^{tr}$ and $\boldsymbol{S}_b^{tr}$ are the within-class and between-class scatter matrices, respectively. Under the above settings, the objective of the LDA model can be interpreted as to find a hyperplane, indicated as $\boldsymbol{w}$, where the ratio between the between-class variance in $\boldsymbol{S}_b^{tr}$ and the within-class variance in $\boldsymbol{S}_w^{tr}$ of the projected data is maximized. Alternatively speaking, the LDA model is trying to maximize the distance between the class means of the N_d classes and at the same time, minimize the variance in each class. Mathematically, this can be described by the maximization of Fisher's criterion [112] as follows:

$$\max_{\boldsymbol{w}} F(\boldsymbol{w}) = \max_{\boldsymbol{w}} \frac{\boldsymbol{w}^T \boldsymbol{S}_b^{tr} \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w^{tr} \boldsymbol{w}} \tag{6.3}$$

For this optimization problem, we can replace the denominator with an equality constraint and get only one solution without losing any generality. Hence, the problem can be reformulated as:

$$\begin{aligned} &\max_{\boldsymbol{w}} \ \boldsymbol{w}^T \boldsymbol{S}_b^{tr} \boldsymbol{w} \\ &\text{s.t.} \ \ \boldsymbol{w}^T \boldsymbol{S}_w^{tr} \boldsymbol{w} = 1 \end{aligned} \tag{6.4}$$

The corresponding Lagrangian function for the above problem can be expressed as:

$$L_{\mathrm{LDA}}(\boldsymbol{w}, \lambda) = \boldsymbol{w}^T \boldsymbol{S}_b^{tr} \boldsymbol{w} - \lambda(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{S}_w^{tr} \boldsymbol{w} - 1) \tag{6.5}$$

where λ is the Lagrangian multiplier associated with the equality constraint in Eq. (6.5). As the scatter matrix $\boldsymbol{S}_b^{tr}$ is positive semidefinite according to Eq. (6.2), this optimization problem is convex, and the global maximum can be reached by equating the derivative of $L_{\mathrm{LDA}}(\boldsymbol{w}, \lambda)$ to zero:

$$\frac{\partial L_{\mathrm{LDA}}(\boldsymbol{w}, \lambda)}{\partial \boldsymbol{w}} = \boldsymbol{S}_b^{tr} \boldsymbol{w} - \lambda \boldsymbol{S}_w^{tr} \boldsymbol{w} = \boldsymbol{0} \tag{6.6}$$

By simple manipulation, Eq. (6.6) can be written as a form of the generalized eigenvalue problem as follows:

$$\boldsymbol{S}_b^{tr} \boldsymbol{w} = \lambda \boldsymbol{S}_w^{tr} \boldsymbol{w} \tag{6.7}$$

Solving the eigenvalue problem, the optimal solution of $\boldsymbol{w} \in R^m$ is the eigenvector corresponding to the largest eigenvalue, i.e., the largest λ. In general, multiple eigenvectors corresponding to the largest few eigenvalues can be selected to formulate a multi-dimensional latent space in which the original data will be projected. Here, we define the letter L as the number of the eigenvectors corresponding to the largest L eigenvalues ($L \leq N_d$). These L eigenvectors $\boldsymbol{w}_l$ ($l = 1, \ldots, L$) form a projection matrix $\boldsymbol{W} \in R^{m \times L}$, which is able to transform (project) the training sample $\boldsymbol{x}_{i,d}^{tr}$ to the latent space where the separation between clusters of classes is maximized and the within variance of each class is minimized. Mathematically, this can be expressed as:

$$\boldsymbol{u}_{i,d}^{tr} = \boldsymbol{W} \boldsymbol{x}_{i,d}^{tr} \tag{6.8}$$

where $\boldsymbol{u}_{i,d}^{tr} \in R^L$ is the latent-space representation of the training sample $\boldsymbol{x}_{i,d}^{tr}$.

After completing the training phase, the LDA model can be used for localizing the areas of possible damage when a testing dataset $\left\{\boldsymbol{x}_{i,d}^{te}\right\}_{i=1;\, d=1}^{N^{te};\, N_d}$ becomes available. Each of the N_d locations (classes) has an equal number of N^{te} testing samples, and we define C_d^{te} as the set of the testing

samples for class d. By projecting the new testing dataset through the previously trained LDA model, the latent-space representation $\boldsymbol{u}_{i,d}^{te}$ of the testing sample $\boldsymbol{x}_{i,d}^{te}$ can be expressed as:

$$\boldsymbol{u}_{i,d}^{te} = \boldsymbol{W}\boldsymbol{x}_{i,d}^{te} \tag{6.9}$$

With regard to the projected training and testing data in the latent space, a proper metric can be employed as a damage index to compute the distance between the projected training and testing data for each of the N_d locations (classes) to quantify the corresponding damage levels. In this work, the Euclidian distance between the class means of the projected training and testing data clusters for each of the N_d classes, is chosen as a damage index, expressed as:

$$E_d = \sqrt{(\overline{\boldsymbol{u}}_d^{tr} - \overline{\boldsymbol{u}}_d^{te})^T(\overline{\boldsymbol{u}}_d^{tr} - \overline{\boldsymbol{u}}_d^{te})} \tag{6.10}$$

where $\overline{\boldsymbol{u}}_d^{tr} = \frac{1}{N^{tr}}\sum_{i=1(i\in C_d^{tr})}^{N^{tr}} \boldsymbol{u}_{i,d}^{tr}$ and $\overline{\boldsymbol{u}}_d^{te} = \frac{1}{N^{te}}\sum_{i=1(i\in C_d^{te})}^{N^{te}} \boldsymbol{u}_{i,d}^{te}$ are the means of the projected training and the testing data clusters of class d, respectively.

In the implementation process of damage localization, the two-sample t-test is firstly performed to determine if the means of the projected training and testing clusters are equal, with respect to each of the N_d classes (locations). Based on a 95% level of statistical significance, a returned p-value of less than 0.05 would demonstrate a significant difference between the means of the two clusters, thus indicating the occurrence of damage at the corresponding location. If there is damage to the monitored structure (at one or more locations), all the distances E_d for $d = 1, \dots, N_d$ will be then computed and compared. The returned maximum E_d indicates the location with the maximum deviation between the 2 clusters, and this can be interpreted as the location where the largest damage has occurred. Specifically, consider a simple situation where the testing data comes from a structure where the stiffness reduction is at a point between two adjacent recording locations, e.g., a column between two adjacent monitored floors of a multi-story shear-

type structural system. In this case, the largest values of E_d will appear in correspondence of the two adjacent monitored floors next to the damage (the stiffness reduction point) (see Section 6.3.1).

A flow chart of the implementation of the above procedure is given in **Figure 6.2**.

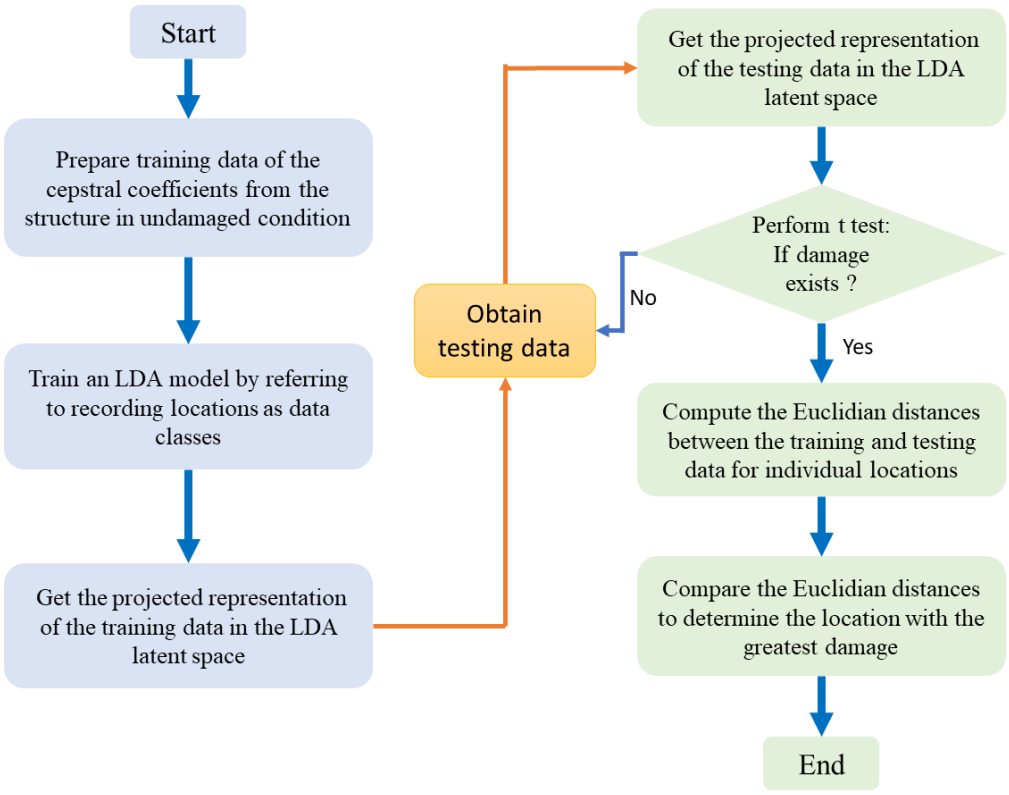

Figure 6.2: A flowchart of the implementation steps for the proposed LDA-based damage localization method.

6.2.3 Damage localization for nonlinear structural systems

As discussed in Section 6.1, real-life structural systems usually experience progressive damage conditions which result in a nonlinear dynamic behavior; this can be generally expressed as a continuously varying process in terms of the structural characteristics and vibration response. Such nonlinear behavior of the damage state is strongly different from the damage scenarios previously discussed in Chapters 5 and 6, where only a constant small stiffness reduction in some

of the elements occurs. Now, instead, we are considering the case where the system can change its structural properties within a given recording. In a nonlinear system, a sudden surge of external excitation can cause the structural system to exhibit nonlinear behavior with damage at one or multiple locations (DOFs), e.g., a multi-story building subjected to an earthquake. In general, when the external excitation returns to normal levels, the entire system may settle and approach a linear state again, but there could be irreversible changes in its structural properties, resulting in completely different dynamic characteristics from the initial undamaged state.

Therefore, a study is conducted to extend the previously proposed data-driven damage localization method to nonlinear systems to identify the location of a damage area if evaluated within a time history. To validate the effectiveness of the method, an analytical modeling strategy is employed to build nonlinear structural systems that can undergo the change in states between linear and nonlinear behavior.

6.2.3.1 Analytical Modeling of Nonlinear Systems

In this section, an analytical modeling approach for simulating the acceleration response of a nonlinear structural system is presented. Let us consider a non-linear N_d DOF model of a general discrete shear-type structure, whose equation of motion can be expressed as:

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{r}(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), t) = \boldsymbol{u}(t) \tag{6.11}$$

where the matrix $\boldsymbol{M} \in \mathbb{R}^{N_d \times N_d}$ represents the system's diagonal mass matrix, consisting of N_d mass elements, while $\boldsymbol{u}(t) \in \mathbb{R}^{N_d}$ represents the external force vector acting on the system. $\boldsymbol{x}(t) \in \mathbb{R}^{N_d}, \dot{\boldsymbol{x}}(t) \in \mathbb{R}^{N_d}$ and $\ddot{\boldsymbol{x}}(t) \in \mathbb{R}^{N_d}$ are namely the displacement, velocity, and acceleration vectors of the system. The term $\boldsymbol{r} \in \mathbb{R}^{N_d}$ represents the restoring force vector that is a function of $\boldsymbol{x}(t)$, $\dot{\boldsymbol{x}}(t)$, and t. The displacement vector $\boldsymbol{x}(t)$ and the restoring force vector $\boldsymbol{r}$ of the N_d-DOF system can be described as:

$$\boldsymbol{x}(t) = [x_1(t), x_2(t), \dots, x_{N_d}(t)] \tag{6.12}$$

$$\boldsymbol{r} = [r_1, r_2, \dots, r_{N_d}] \tag{6.13}$$

where $x_d(t)$ $(d = 1, \dots, N_d)$ represents the nodal displacement at the d^{th} mass relative to the base of the structural system, and r_d represents the nonlinear restoring force between the d^{th} and $(d-1)^{\text{th}}$ masses. In this study, the Bouc-Wen (BW) model, originally proposed by Bouc [113] and Wen [114], is used to represent a smooth hysteretic restoring force of the system: This BW model can be generalized by adding a linear viscous damping parameter c_d and a stiffness parameter l_d related to the cubic power of the displacement. The restoring force r_d of this generalized BW model can be then expressed as:

$$\begin{aligned} \dot{r}_d = {} & c_d(\ddot{x}_d - \ddot{x}_{d-1}) + k_d(\dot{x}_d - \dot{x}_{d-1}) + l_d[3(x_d - x_{d-1})^2(\dot{x}_d - \dot{x}_{d-1})] \\ & + b_d|\dot{x}_d - \dot{x}_{d-1}||r_d|^{power-1} r_d + e_d(\dot{x}_d - \dot{x}_{d-1})|r_d|^{power} \end{aligned} \tag{6.14}$$

where $\ddot{x}_{d-1} = \dot{x}_{d-1} = x_{d-1} = 0$ for the case of $d = 1$, i.e., the bottom mass of the system. The structural parameters c_d, k_d, l_d, b_d, e_d, and *power* are selected by the user in accordance with the type of nonlinearities to be represented in the model. Once the external force $\boldsymbol{u}(t)$ is known, the acceleration response of the nonlinear system can be simulated by solving Eq. (6.11) using, for example, a third-order Predictor–Corrector integration scheme [115].

The time histories of the structural acceleration can be obtained as described below:

1) Based on a recursive form with successive forward time steps, an observation matrix with respect to the d^{th} DOF at time step k, denoted as $\boldsymbol{\psi}_d(k) \in \mathbb{R}^{3\times 5}$, can be firstly formed according to the third-order setting of the Predictor-Corrector integration and to the polynomial in Eq. (6.14) (only the first 2 linear components in Eq. (6.14) are shown for simplicity.):

$$\boldsymbol{\psi}_d(k) = \begin{bmatrix} \ddot{x}_d(k) - \ddot{x}_{d-1}(k), & \dot{x}_d(k) - \dot{x}_{d-1}(k), & \dots, & \dots, & \dots \\ \ddot{x}_d(k-1) - \ddot{x}_{d-1}(k-1), & \dot{x}_d(k-1) - \dot{x}_{d-1}(k-1), & \dots, & \dots, & \dots \\ \ddot{x}_d(k-2) - \ddot{x}_{d-1}(k-2), & \dot{x}_d(k-2) - \dot{x}_{d-1}(k-2), & \dots, & \dots, & \dots \end{bmatrix} \tag{6.15}$$

and the corresponding parameter vector for the matrix $\boldsymbol{\psi}_d(k)$, denoted as $\boldsymbol{\theta}_d(k) \in \mathbb{R}^5$, can be expressed as:

$$\boldsymbol{\theta}_d(k) = [c_d, k_d, l_d, b_d, e_d]^T \tag{6.16}$$

2) Then, the Predictor part of the Predictor-Corrector method can be implemented to get an estimate of the predicted restoring force $\hat{r}_d(k+1)$ with respect to the d^{th} DOF for time step $k+1$, expressed as:

$$\hat{r}_d(k+1) = r_d(k) + h\boldsymbol{c}_c\boldsymbol{\psi}_d(k)\boldsymbol{\theta}_d(k) \tag{6.17}$$

where h represents the sampling time, and the third-order coefficient vector $\boldsymbol{c}_c$ is defined as:

$$\boldsymbol{c}_c = \frac{1}{12}[5,8,-1] \tag{6.18}$$

3) The corresponding predicted acceleration $\hat{\ddot{x}}_d(k+1)$ can be then easily achieved by substituting the predicted restoring force $\hat{r}_d(k+1)$ into Eq. (6.11), and the corresponding predicted velocity $\hat{\dot{x}}_d(k+1)$ and displacement $\hat{x}_d(k+1)$ can be obtained by simply using a numerical integration method (e.g., the Newmark-beta method).

4) The achieved $\hat{\ddot{x}}_d(k+1)$, $\hat{\dot{x}}_d(k+1)$, and $\hat{x}_d(k+1)$ can be then substituted into Eq. (6.14) to get the corresponding predicted first-order derivative of the restoring force, denoted as $\hat{\dot{r}}_d(k+1)$.

5) Next, the Corrected part of the Predictor-Corrector method is implemented to get the corrected restoring force $r_d(k+1)$, with respect to the d^{th} DOF at time step $k+1$, as follows:

$$r_d(k+1) = r_d(k) + h\boldsymbol{c}_c[\hat{\dot{r}}_d(k+1), \dot{r}_d(k), \dot{r}_d(k-1)] \tag{6.19}$$

This corrected restoring force $r_d(k+1)$ can be substituted into Eq. (6.11) to obtain the corresponding corrected acceleration $\ddot{x}_d(k+1)$, along with the numerical integration for the corresponding corrected velocity $\dot{x}_d(k+1)$ and displacement $x_d(k+1)$.

The above process is then repeated to simulate the acceleration response of the nonlinear system for the entire duration of the external force. Afterwards, the cepstral coefficients of the simulated acceleration response of the nonlinear system can be numerically extracted by the same approach of digital signal processing as introduced in Chapters 4 and 5.

6.2.3.2 Implementation of nonlinear system damage localization

The damage identification and localization for a nonlinear system can be performed in a similar way as done for linear systems. A model is first built and trained with the data obtained from the system in its undamaged state (still in a linear stage). Then, when processing new testing data, if the system starts exhibiting a nonlinear behavior, the trained model can identify potential local anomalies (damage) at various recording locations.

When working on case studies of linear structural systems (as presented in Chapters 4 – 5 and Section 6.3.1), the damaged states of the linear systems are simulated by setting various stiffness reductions at different locations (DOFs) of the system. These stiffness reduction settings are made prior to each simulation and do not change during the simulation. On the contrary, to simulate a progressive damage condition, a nonlinear system shows continuous variation of its parameters within a single recording, mainly linked to dramatic variation of the external excitation.

According to the above mindset, a general implementation process of the data-driven damage localization for nonlinear systems can be described as follows: 1) The structural acceleration response data of the monitored nonlinear system in an undamaged state are first obtained and, if necessary, appropriate framing operations are performed. 2) The acceleration response data are

processed by the signal processing approach introduced in Section 4.2.1 to extract the corresponding cepstral coefficients, which will serve as the training dataset. 3) A proper data-driven model, e.g., the proposed New Generalized Auto-Encoder (NGAE) in Chapter 4 or the proposed LDA model in Section 6.2.2, is built and trained with the training dataset. 4) When new data is obtained from the system in an unknown state, these data will go through the same framing operations and cepstral coefficient extraction. 5) The trained model is finally used to locate the potential damage of the nonlinear system over the testing data, via the damage index of the Euclidian distance introduced in Section 6.2.2.

6.3 Results

In this section, two sets of results are discussed, corresponding to the problem of structural damage localization for linear and nonlinear systems, respectively. In Section 6.3.1, the results of a numerical case study (modeling and analysis of an 8 DOF shear type structural system) are presented for validating the effectiveness of the proposed damage localization method for linear systems. In Section 6.3.2, a single DOF nonlinear system and a 4 DOF nonlinear shear-type system, built upon the generalized BW model (Section 6.2.3.2), are discussed to validate the proposed methodology for damage localization in nonlinear systems.

6.3.1 Results of structural damage localization for a linear 8 DOF system

In this case study, the same lumped mass model of the 8 DOF shear-type system, introduced in Section 4.3.1 and shown in **Figure 4.4,** was used to validate the proposed damage localization method for linear systems. As a reminder, the baseline conditions of the system are: The baseline stiffness of the 8 vertical elements is set to the same value as $k_1^0 = k_2^0 = \cdots = k_8^0 = k^0 =$

$25{,}000\ \mathrm{N/m}$, and each mass is equal to $m_d = 1\ kg$ $(d = 1, \ldots 8)$. The assumption of modal damping is used, with a damping factor of $\xi = 1\%$ for each of the 8 vibration modes. Seventeen different scenarios were simulated as shown in **Table 6.1**, for different operational and damage conditions; damage was introduced by changing the baseline stiffnesses of some elements. The first 9 scenarios in **Table 6.1** represent the undamaged states of the structural system under a variety of different environmental conditions, while the remaining 8 scenarios (scenarios 10 – 17) represent the cases where damage is set to occur separately at each of the 8 DOFs with the inter-story stiffness at each DOF reduced by 25%. Note that in this case, the damage-location setting for a DOF d indicates that the stiffness reduction is at the columns between the DOF d and the DOF $d-1$. For the case of $d = 1$, it means the stiffness reduction is at the location between the bottom DOF and the ground.

For each scenario, the excitation is provided by 8 external forces applied at the 8 masses. The 8 external forces are modeled as 8 zero-mean Gaussian white noises with the zero-order-hold (ZOH) assumption and with equal magnitude of 100 N. Each realization of the force has a duration of 500 seconds and it is sampled at 200 Hz. The generated acceleration responses at the 8 DOFs are then corrupted by a 10% RMS Gaussian white noise to simulate measurement error. In this case study, 100 realizations of acceleration responses for each of the 9 undamaged scenarios (scenarios 1 – 9) were simulated, for a total of 900 sequences of the acceleration cepstral coefficients extracted at each DOF d $(d = 1, \ldots, 8)$. All these data were then collected together to form the training dataset $\left\{\boldsymbol{x}_{i,d}^{tr}\right\}_{i=1;\ d=1}^{N^{tr};\ N_d}$ with $N^{tr} = 900$ and $N_d = 8$. The testing data consist of the cepstral coefficients extracted from 50 new realizations of the acceleration responses for each of the 17 scenarios. Those produced from scenarios 1 – 9 are collected together to form one "undamaged" testing dataset $\left\{\boldsymbol{x}_{i,d}^{te}\right\}_{i=1;\ d=1}^{N_u^{te};\ N_d}$, with $N_u^{te} = 450$ and $N_d = 8$, while the remaining

ones (scenarios 10 – 17) form 8 individual "damaged" testing datasets $\{\boldsymbol{x}_{i,d}^{te}\}_{i=1;\,d=1}^{N_s^{te};\,N_d}$ for $s = 1, \ldots, 8$, with $N_1^{te} = N_2^{te} = \cdots = N_8^{te} = 50$ and $N_d = 8$, where s represents the index for the 8 damage scenarios.

Table 6.1: Considered undamaged and damaged scenarios of the 8 DOF shear-type system.

Scenario	Condition	Types of anomalies
1	Undamaged	Baseline scenario
2	Undamaged	$k_d = 0.98k^0$ for $d = 5, 6, 7, 8$
3	Undamaged	$k_d = 0.99k^0$ for $d = 5, 6, 7, 8$
4	Undamaged	$k_d = 1.01k^0$ for $d = 5, 6, 7, 8$
5	Undamaged	$k_d = 1.02k^0$ for $d = 5, 6, 7, 8$
6	Undamaged	$k_d = 0.98k^0$ for $d = 1, 2, 3, 4$
7	Undamaged	$k_d = 0.99k^0$ for $d = 1, 2, 3, 4$
8	Undamaged	$k_d = 1.01k^0$ for $d = 1, 2, 3, 4$
9	Undamaged	$k_d = 1.02k^0$ for $d = 1, 2, 3, 4$
10	Damaged	$k_d = 0.75k^0$ for $d = 1$
11	Damaged	$k_d = 0.75k^0$ for $d = 2$
12	Damaged	$k_d = 0.75k^0$ for $d = 3$
13	Damaged	$k_d = 0.75k^0$ for $d = 4$
14	Damaged	$k_d = 0.75k^0$ for $d = 5$
15	Damaged	$k_d = 0.75k^0$ for $d = 6$
16	Damaged	$k_d = 0.75k^0$ for $d = 7$
17	Damaged	$k_d = 0.75k^0$ for $d = 8$

As introduced in Section 6.2.2, an LDA model was built and trained upon the training data, representative of the undamaged conditions, where the cepstral coefficients of the 8 DOFs were treated as 8 different classes. Consequently, in the latent space of the LDA model, the projected cepstral coefficients of the training data are separated as 8 individual clusters according to the 8 classes (DOFs), as shown in **Figure 6.3 (a)**. Next, this trained LDA model was used to project the cepstral coefficients of the testing data into the same latent space to identify the potential deviation pattern from the distributions of the training data to the testing ones for damage localization. To visualize such deviation, the scatter plots of the latent-space representation, in terms of the training data and the undamaged testing data (scenarios 1 – 9), as well as the training data and the testing

ones of the 2nd damage scenario (scenario 11), are shown in **Figure 6.3 (b)** and **(c)**, respectively. Two important observations can be reached by looking at the scatter plots: 1) In **Figure 6.3 (b)** as expected, the testing data points corresponding to the undamaged conditions perfectly overlap with the training ones for all the 8 DOFs. 2) In **Figure 6.3 (c)**, the distributions of the testing data (scenario 11) of DOF 1 and DOF 2 show the largest 2 deviations from the corresponding distributions of the training data, with the 2 deviations marked by 2 black arrows, while the deviations for the other DOFs are minimal. Such scatter distribution results correctly reflect the fact that the damage location, i.e., the stiffness reduction, is at the columns between DOF 1 and DOF 2.

To quantify and summarize these deviations due to the damage, the Euclidian distance (as introduced in Section 6.2.2) was employed to compute the distances between the training data of the undamaged scenarios and the testing ones of every damaged scenario, for each of the 8 DOFs individually. In this case, the Euclidian distance was used to measure the distances between the centers of the first 2 projected cepstral coefficients (i.e., the 2 components of the LDA latent space linked to the largest 2 ratios of the between- and within-class variances) for the undamaged and the different damaged scenarios. The computed results are summarized as a bar chart shown in **Figure 6.4**. It can be easily found that for each damage case, the maximum 2 Euclidean distances always occur at the 2 DOFs adjacent to the location of the stiffness reduction. These results demonstrate the effectiveness of the proposed LDA-based modeling method in indicating the location with the greatest damage in a linear shear-type system, for the case when there is a only single location of stiffness reduction. In future work, further exploration can be carried out to verify the effectiveness of this approach in damage scenarios with multiple stiffness reduction locations.

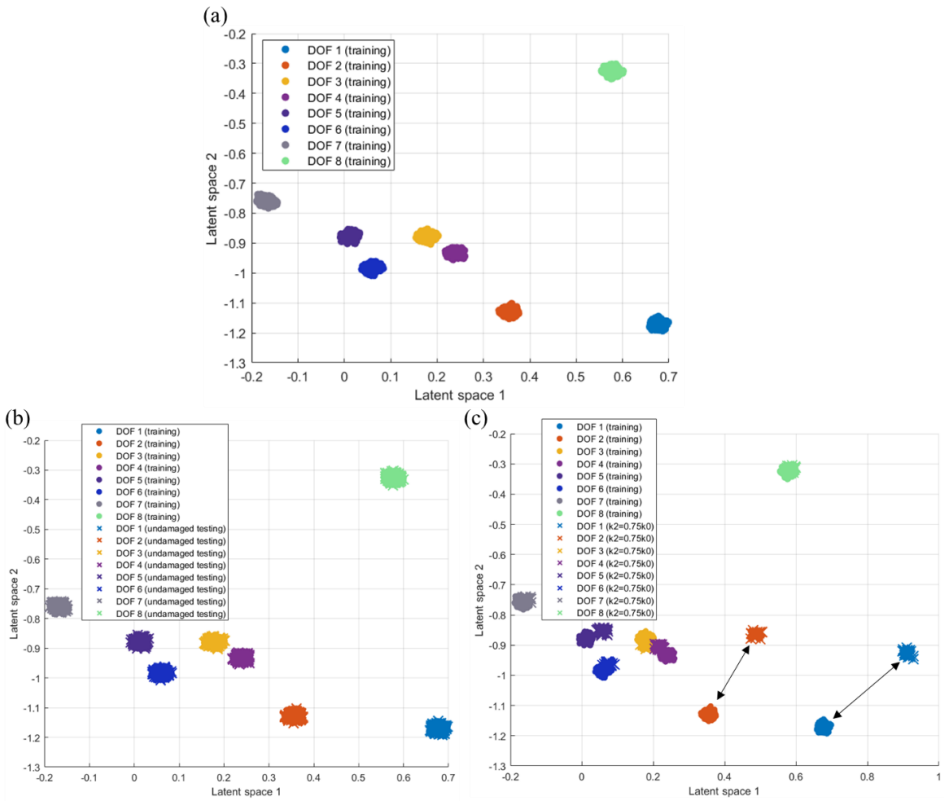

Figure 6.3: The distributions of the projected cepstral coefficients from the 8 DOFs of the system. (a) Results of only training data. (b) Results of the training data and the undamaged testing data (scenarios 1 – 9). (c) The results of training data and the testing data of scenario 11.

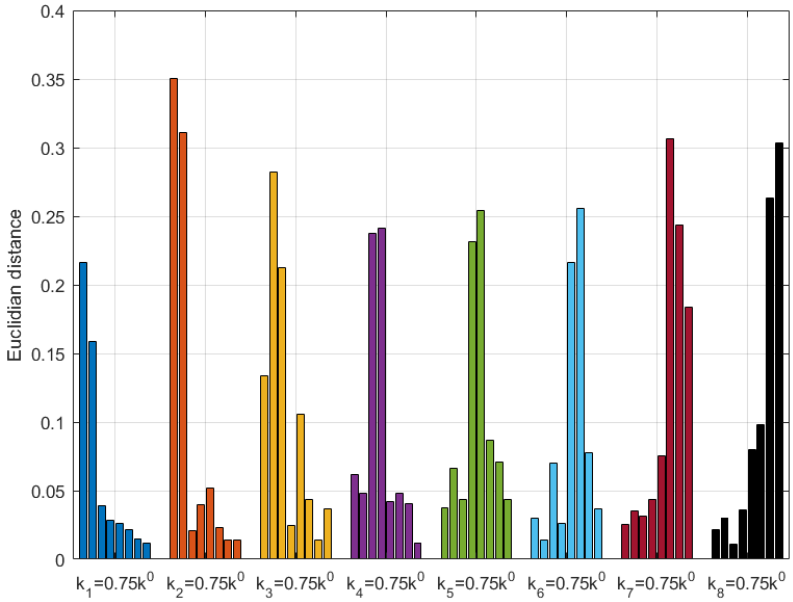

Figure 6.4: The Euclidian distance between the centers of the first 2 projected cepstral coefficients of the training data (scenarios 1 – 9) and of the testing data for each of the damage scenarios (scenarios 10 – 17).

6.3.2 Results of structural damage localization for nonlinear systems

6.3.2.1 Nonlinear behavior detection of a single DOF system

For the problem of the damage identification and localization in nonlinear systems, we started with analyzing the case of a Single-DOF (SDOF) nonlinear system. This SDOF nonlinear system was simulated using the generalized BW model (Section 6.3.2). The following values of the structural parameters, as presented in Eq. (6.14), were used as baseline settings for the SDOF system: $m = 1\ \text{kg},\ c = 0\ \text{N}\cdot\text{s/m},\ k = 5\ \text{N/m},\ l = 0,\ b = -0.1,\ e = -1,\ \text{power} = 2$ (l, b, e, and $power$ are dimensionless parameters corresponding to the nonlinear restoring force term of the system.). As previously discussed, the process for detecting the occurrence of nonlinear behavior within a system can be summarized as follows: A data-driven model is first built based on the training data from the system in its undamaged state (i.e., linear behavior state). This is followed by an unsupervised-learning-based strategy to identify the abnormality in the testing data obtained from the system with potential nonlinear behavior due to some extreme conditions, e.g., a substantial increase in the magnitude of excitation forces.

The training data of the SDOF system's response were simulated using a baseline excitation force of small magnitude. By trial-and-error calibration and the rule of thumb discussed in [115], this baseline excitation for simulating the training data was set equal to a Random Gaussian Signal (RGS) with zero mean and a standard deviation of 0.1 N. Such a setting is to create a condition where the system, subjected to such level of excitation, is able to exhibit stable linear behavior throughout the entire time duration of each realization. In this case, the duration of each realization was set equal to 40 seconds with a sampling interval of 0.01 seconds. To get rid of the effects caused by initial conditions, the first 10 seconds of data of each realization were removed.

In order to generate testing response data that are the results of the nonlinear behavior of this SDOF system, in each simulation, the magnitude of excitation applied to the system was set to vary in three different phases. The time duration of each realization was set equal to 100 seconds with the sampling period of 0.01 second. The corresponding excitation, set to be an RGS, was divided into 3 segments: 0 – 40 seconds, 40 – 70 seconds and 70 – 100 seconds. For the first segment (0 – 40 seconds), the excitation had the same magnitude (zero mean and the standard deviation of 0.1 N) as the baseline setting used in generating the training data. For the second segment (40 – 70 seconds), the magnitude of the excitation was substantially increased so to have a standard deviation of 4 N: During this potion of the record, the magnified force is expected to push the SDOF into the nonlinear range and show a nonlinear behavior. Finally, for the last segment of 70 – 100 seconds, the excitation was set back to the initial values (zero mean and standard deviation of 0.1 N) so that the system can "settle down" in a new damaged condition. As with the training data, the first 10 seconds of the response of every realization of the testing data was removed to clear the effect of initial conditions. Thus, the length of the entire response for each realization becomes 90 seconds, and the length of the varying excitation for each of the three segments is 30 seconds.

To have an intuitive sense of the difference between the linear and nonlinear behavior of the SDOF system, the relationship between the SDOF system displacement and the restoring force, in terms of the simulated training and testing data, is visualized in **Figure 6.5 (a)** and **(b)**, respectively. Note that **Figure 6.5 (a)** presents the result of one sample realization of the training data, where the response is 30-second length, while **Figure 6.5 (b)** shows the results of one 90-second sample realization from the testing data. One can observe that the displacement and the restoring force for the training dataset show a linear relationship, with a clear straight line between

the two variables as shown in **(a)**, indicating that the system is in an undamaged state with almost completely linear behavior throughout the time duration. For the testing data results in **(b)**, the three 30-second parts of the response produce three different restoring force – displacement relationships: 1) During the first 30 seconds, since the excitation in this phase has the same small magnitude as that of the baseline setting, the behavior of the system remains linear, with the displacement and the restoring force showing a linear relationship (shown as the thick, short blue line). 2) During the second 30 seconds, as the magnitude of the excitation has largely increased, the system starts behaving nonlinearly, with the relationship between the two variables showing a clear hysteretic behavior as shown by the orange line. 3) After moving into the last 30 seconds, the system largely settles down as the excitation magnitude has been tuned back to the original values. However, due to the irreversible stiffness change caused by the damage in the second phase, it cannot recover back to the original linear state, and the two variables thus still show a slight nonlinear relationship, as shown by the green line.

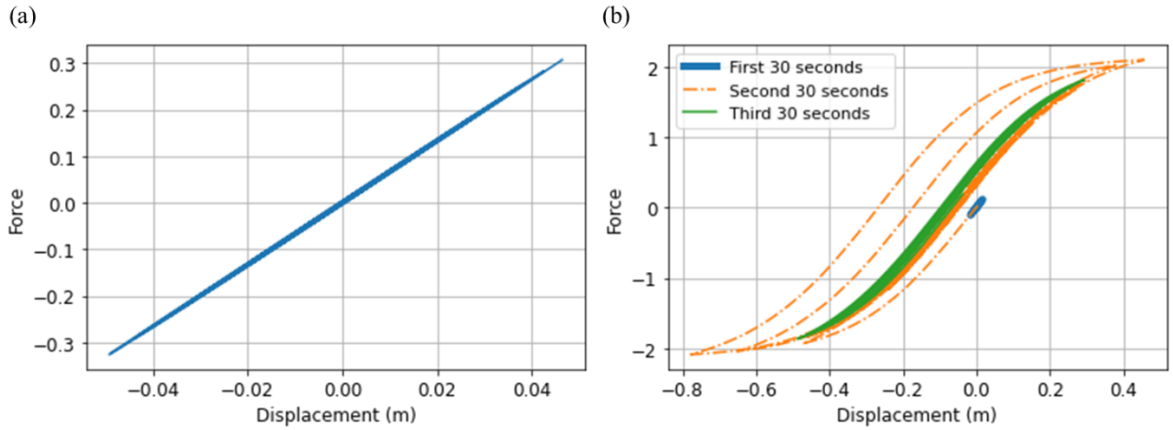

Figure 6.5: Relationship between the displacement and the restoring force of the SDOF system, (a) results of the training data, and (b) results of the testing data.

As discussed earlier, the goal is to build a data-driven model that can accurately and timely identify the occurrence of nonlinear behavior of the system. For this purpose, the modeling was

again conducted using the cepstral coefficients of the acceleration response of the system. For the training dataset, 500 realizations of the 30-second acceleration response with the baseline settings of the system's parameters and the RGS excitation were used, from which the corresponding cepstral coefficients were extracted. For testing, 100 realizations of the 90-second acceleration response, under the condition of the three-stage excitation, were simulated first. For each realization, the cepstral coefficient extraction was performed on the acceleration data of the three phases separately. Accordingly, for each of the 100 realizations, three cepstral coefficient vectors, representing the three different stages (i.e., the linear – nonlinear – linear stages) of the system, were obtained.

In this case, a New Generalized Auto-Encoder (NGAE) (Section 4.2.2) was built and trained based on the training dataset of the cepstral coefficient vectors: using the NRMSE and the SDR metrices in Section 4.2.3, it is possible to see that the cepstral coefficients follow a robust Gaussian training distribution for the values of the two metrices, and this distribution is representative of the undamaged linear state of the system. Subsequently, the trained NGAE was used to identify the nonlinear behavior present in the testing data. In this case, three distributions of the two metrices were generated and compared with the training distributions (**Figure 6.6)**. It can be observed that the testing distribution of the metrices for the first 30 seconds, corresponding to the same magnitude of the applied excitation as that of the baseline setting, completely overlaps the training distribution, confirming that during the first 30 seconds, the system maintains a linear behavior and is in its undamaged state. Moving to the next 30 seconds, the corresponding testing distribution shows a significant deviation pattern from the training distribution as expected. This indicates that the system, exhibiting a nonlinear behavior, suffers damage and the NGAE can successfully identify such a mechanical change of the system. For the last 30 seconds, when the excitation is

adjusted back to the original magnitude, the corresponding testing distribution greatly differ from the previous two, indicating that the system is now in a completely new state and cannot operate as in the beginning of the record due to the irreversible changes in its structural properties, caused by the damage during the second phase.

To characterize the changing behavior of this nonlinear SDOF system, the varying distributions of the cepstral coefficients produced under this three-stage excitation condition were further analyzed in a tracer manner by analyzing their evolution in time. We first randomly selected one realization of the response in the testing dataset, and then performed framing operations over the entire simulation duration, with the cepstral coefficient extraction on each frame. For these cepstral coefficients, a Principal Component Analysis (PCA) was subsequently performed to highlight how their distributions vary as result of the varying three-stage excitation. **Figure 6.7** shows the variation of the first 2 principal components of the PCA latent space, with the results generated by the three phases of excitation represented by three colors (blue, orange, and red). The black line with the arrow indicates the moving-average of these 2 components, starting from the center of the cluster (the blue one) corresponding to the first 30-second response. This tracking result clearly shows the significant difference in the system behavior from a stable linear state to varying nonlinear states (from the blue cluster to the orange one, then to the green one), and indicates that the structural properties of the system constantly change after suffering damage.

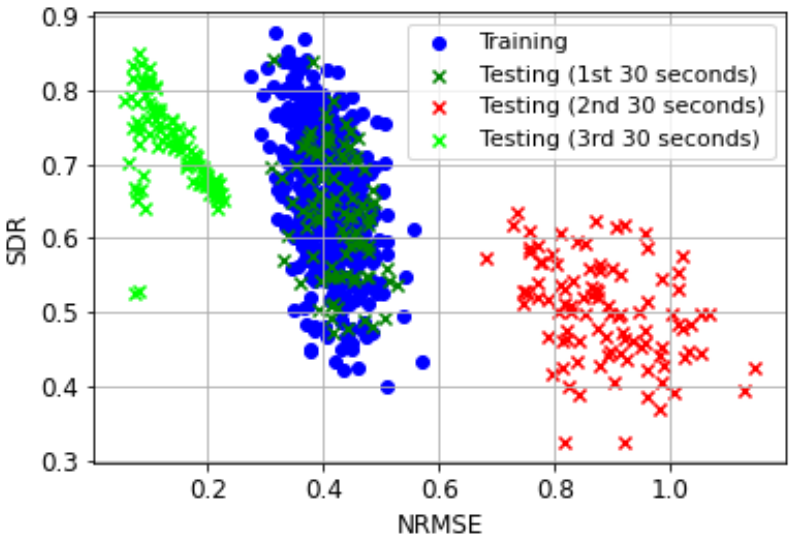

Figure 6.6: The distributions of the NRMSE and the SDR with respect to the training and the testing datasets, where the results of the test data are presented for each of the three phases.

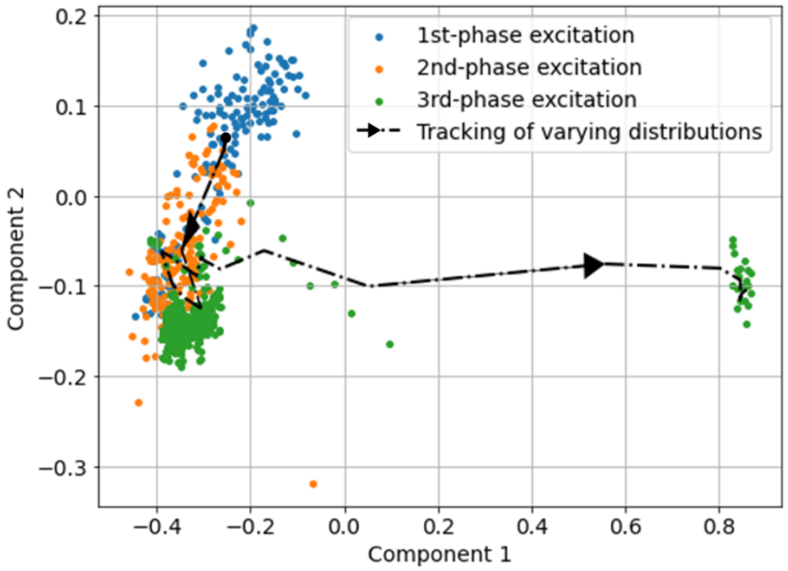

Figure 6.7: Tracking for the varying distribution of the projected cepstral coefficients.

6.3.2.2 Damage localization in a 4 DOF nonlinear system

After testing the cepstral coefficient-based damage assessment strategy for a SDOF nonlinear system, the proposed damage localization method (Sections 6.2.2-6.2.3) was tested on a nonlinear Multi-DOF (MDOF) system. In this case, a 4-DOF shear-type structural system with inter-story

nonlinear elements represented as the generalized BW model (Section 6.2.3.1) was analyzed (**Figure 6.8)**.

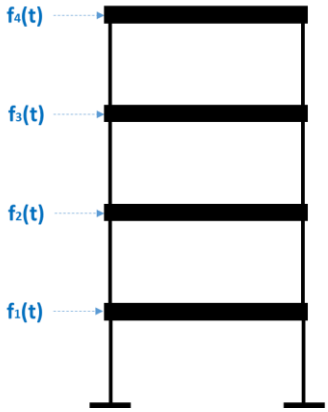

Figure 6.8: A 4-DOF shear type structural system.

To simulate the normal operational state of the 4-DOF system, based on the results shown in [54, 115], the following parameters were considered as baseline settings: $m_d = 1000$ kg, $c_d = 51.8\ \text{kN} \cdot \text{s/m}$, $k_d = 2200$ kN/m, $l_d = 0$, $b_d = -200$, $e_d = -100$ for $d = 1, \ldots, 4$, where d represents the d^{th} DOF (from bottom to top). Four RGS-based excitation forces, all with a magnitude of zero mean and a standard deviation of 0.1 kN, were applied at the 4 masses of the system, and the output consisted of the time histories of the acceleration of the 4 DOFs. Note that these baseline settings of the structural parameters and applied excitations are set so to simulate an undamaged state of the system with little to no nonlinear behavior since the set excitation magnitude is sufficiently small so to avoid nonlinearities (as shown in **Figure 6.9 (a)**). With these baseline settings, 100 realizations of the acceleration response of the 4 DOF system were simulated, and the cepstral coefficients were extracted from these records to form the training dataset.

To induce a nonlinear behavior in the system, the magnitude of the excitation acting on the bottom DOF ($\text{d} = 1$) of the system was increased to a standard deviation of 10 kN, while the other

3 were maintained with a standard deviation of 0.1 kN. Under the action of these new excitations, the 4 DOF system is expected to exhibit nonlinear behavior at all locations, with the bottom inter-story element being the one most affected by the large excitation (as shown in **Figure 6.9 (b)**). With this new excitation setup, 50 realizations of the acceleration response were then simulated, and the corresponding cepstral coefficient were extracted. This dataset represents the testing dataset.

Figure 6.10 presents an intuitive comparison of the trends of the cepstral coefficients produced in the scenario of the baseline excitations at 4 DOFs **(a)** and in the damage scenario of the large excitation at DOF 1 **(b)**. It can be observed that, after a substantial increase in the magnitude of the excitation at DOF 1, the trends of the cepstral coefficients extracted from the acceleration of DOF 1 show a significant discrepancy between the two different excitation conditions, where the trend of the DOF 1's cepstral coefficients generated by the large excitation becomes highly flat (**Figure 6.10 (b)**). In contrast, the cepstral coefficients of the other 3 DOFs produce a much smaller level of the discrepancy, probably due to their milder nonlinearities.

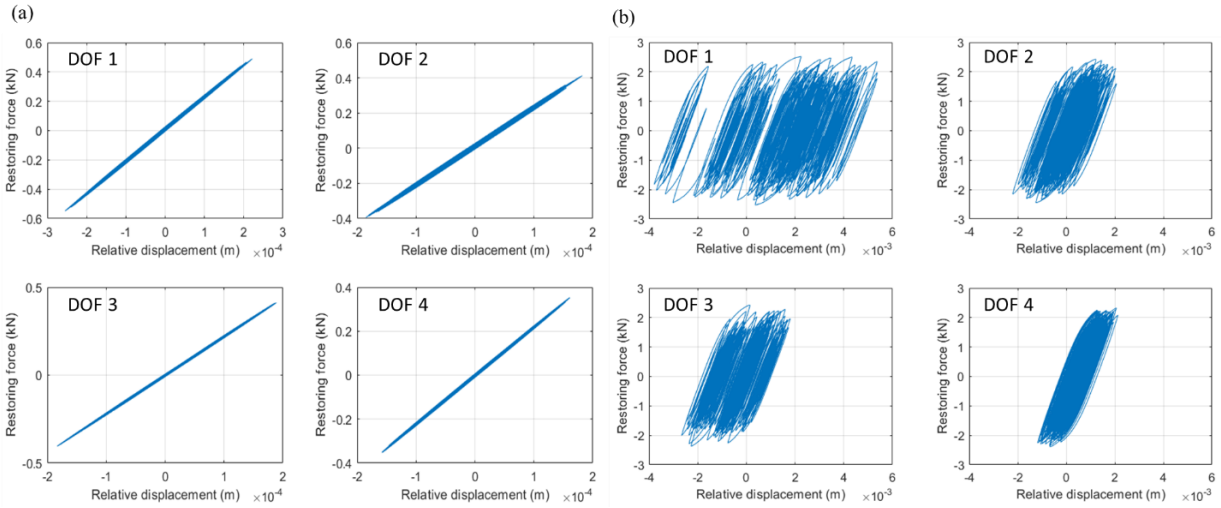

Figure 6.9: Relationships between the relative displacements and restoring forces for each of the 4 DOFs. (a) The undamaged scenario with the baseline excitation condition. (b) The damage scenario with the large excitation at the DOF 1.

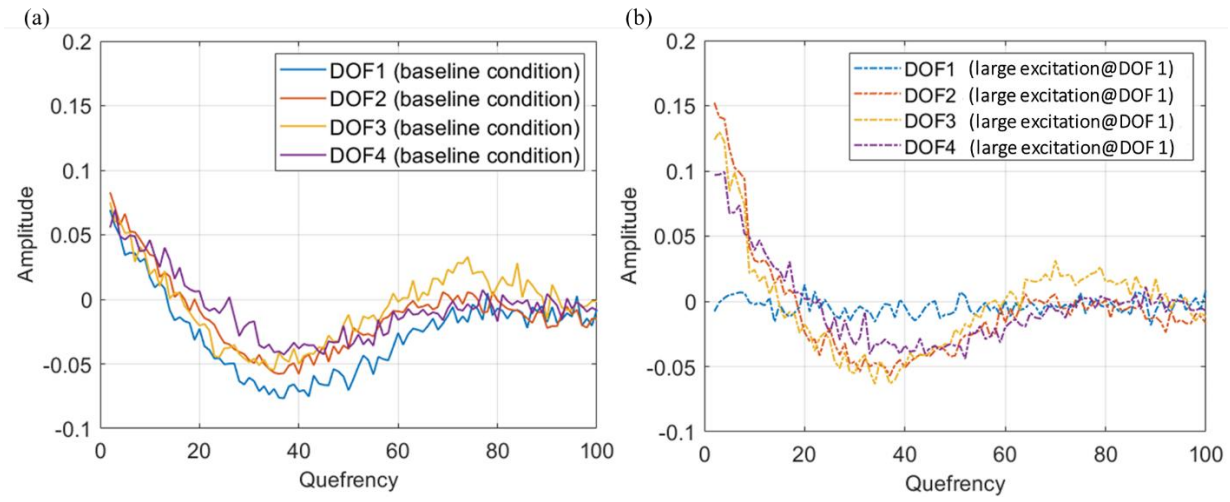

Figure 6.10: A comparison of the cepstral coefficients between the two excitation conditions.

To localize the damaged element, an LDA model was built and trained by using the cepstral coefficients of the training dataset, where the cepstral coefficients were labeled as 4 individual classes according to their locations, i.e., DOFs 1 – 4. The latent-space distributions of the coefficients from the training data in this case are presented in **Figure 6.11 (a)**. It can be observed that there are 4 clusters clearly separated, in terms of the first two principal components of the LDA models, corresponding to the 4 DOFs of the system. When this LDA model is used to project the coefficients of the testing data into the same latent space, it is apparent that the occurrence of damage induces some scattering in the projected coefficients. **Figure 6.11 (b)** presents the distributions of the projected cepstral coefficients of this damage scenario, as well as the results of the training data for comparison. It can be seen that the distribution of values corresponding to DOF 1 shows the largest deviation spreading in comparison with the other three DOFs, correctly indicating that the first inter-story element suffers the largest nonlinearity.

To quantitatively verify the effectiveness of the proposed damage localization method, three more testing scenarios were simulated, i.e., the large excitation is applied to DOF 2, DOF 3 and DOF 4, respectively, to form 3 additional testing datasets (thus a total of 4). The previously trained

LDA model was then used to perform the cepstral coefficient projection onto the same latent space with respect to each of the 4 testing datasets. The Euclidian distances (Section 6.2.2) between the cluster means of the projected coefficients of the baseline condition (the training data) and of the 4 testing scenarios (the testing data), for the 4 DOFs respectively, were used to quantify the deviations of distributions for localizing the structural damage induced by the nonlinear behavior. **Figure 6.12** presents the Euclidian distance between the cluster means for the 4 testing scenarios. Two important points can be reached here: 1) Regardless of where the large excitation is applied to the system, the Euclidean distances of all 4 DOFs always show non-zero values, indicating that the entire system is damaged in all the 4 scenarios. 2) It can be easily observed that the maximum Euclidean distance always appears at the location where the large excitation is applied. Consequently, these results demonstrate that the proposed LDA-based structural damage localization method is able to correctly indicate the location of the inter-story element with the greatest level of damage.

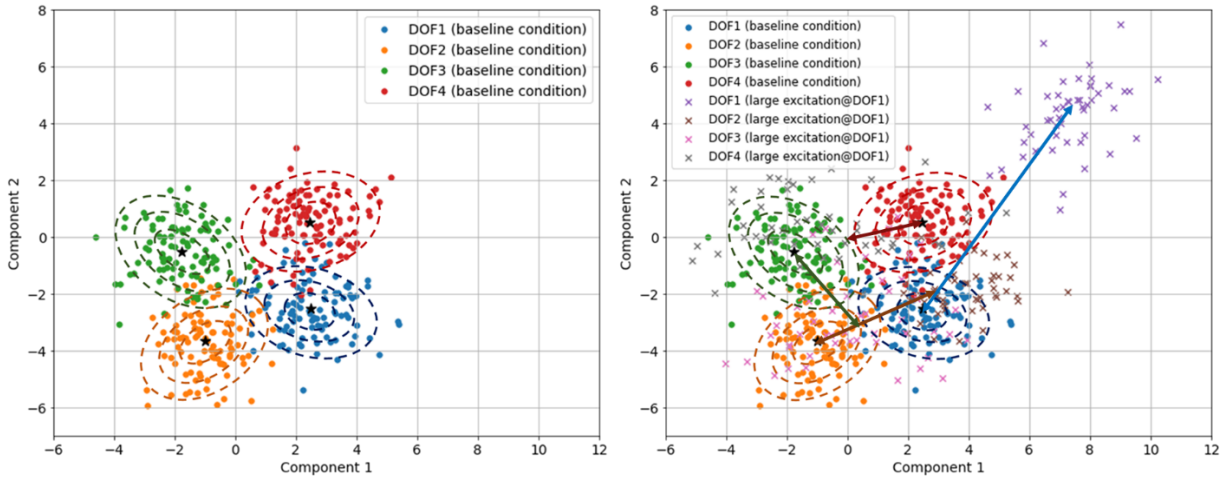

Figure 6.11: The distributions of the projected cepstral coefficients in the first two components of the LDA model. (a) The results of only the training data. (b) The results of the training data and the testing data for the scenario of damage at DOF 1. The two-way arrows indicate the deviation distance from the cluster means of the baseline condition in the 2-D latent space.

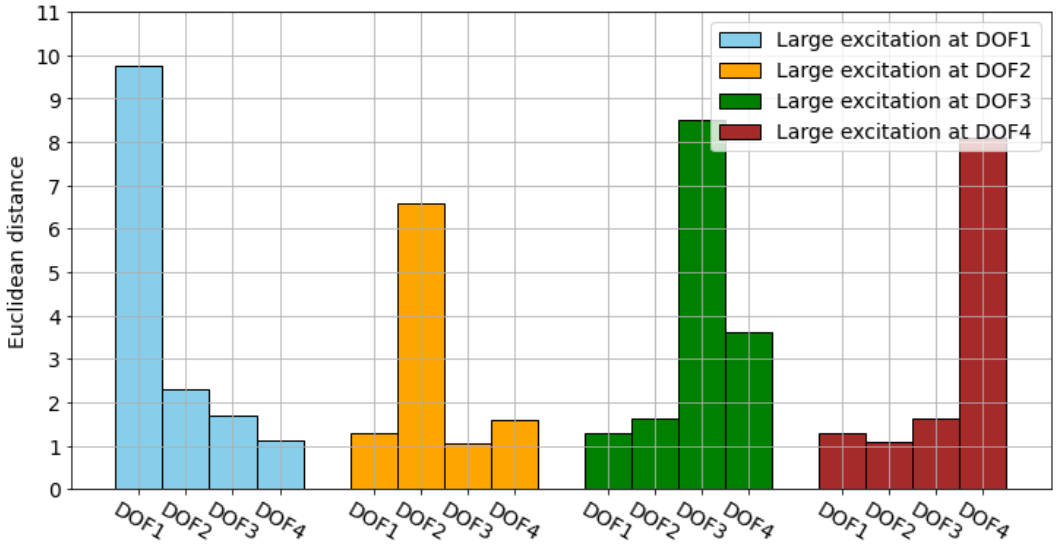

Figure 6.12: The Euclidian distances of the cluster means between the undamaged scenario and each of the 4 damage scenarios.

6.4 Conclusions

In this chapter, a new data-driven modeling methodology to address structural damage localization problems is proposed. This methodology is based on the Linear Discriminant Analysis (LDA) and uses the local characteristics embedded in the cepstral coefficients of the structural acceleration response. The proposed LDA-based model is able to highlight the structural local characteristics embedded in the cepstral coefficients through its ability to maximize the separation of categorical data in the LDA latent space. The projected cepstral coefficients can help perform the localization of structural damage by using the Euclidean distance between the means of distributions as damage index to quantify the damage levels at different monitoring locations. The effectiveness of the proposed method has been verified by two case studies of a linear and a nonlinear structural system, and the findings can be summarized as follows.

1) For linear systems with data corresponding to 2 different conditions (undamaged and damaged), the LDA model, trained only with data obtained from the system in undamaged

conditions, can successfully point out the locations of stiffness reduction in an entire unsupervised-learning manner.

2) For nonlinear systems, when the LDA model is trained with data obtained from low-level vibrations (e.g., indicative of a linear behavior), such a model can correctly determine the locations where inter-story elements suffer large nonlinearities. Looking at the values of the Euclidean distances between clusters also provides an indication of how severe the nonlinearities are over the entire structure.

It would be interesting to test the proposed damage localization strategy with data collected from real structures and to see its performance on more complex damage scenarios.

Chapter 7. Conclusions and future directions

7.1 Conclusions

In this dissertation, state-of-the-art data-driven methods have been explored for two areas in civil engineering, namely Residential Electrical Modeling (REM) and Structural Health Monitoring (SHM). More specifically, this dissertation presents detailed descriptions of the systematic development of novel data-driven methods to five typical problems in these two areas, namely, short-term electricity load forecasting and electricity load peak forecasting in REM, and structural damage detection, classification, and localization in SHM.

In Chapter 2, the short-term electricity load forecasting in residential buildings, an important problem in REM, was fully investigated and a novel Recurrent Neural Network (RNN)-based model was proposed. This model is an integration of a modified Convolutional Long Short-Term Memory (ConvLSTM) neural network with selected autoregressive features, termed as a CLSAF model, which is aimed to improve single-step-ahead electricity load forecasting for three spatial granularities: 1) apartment, 2) floor, and 3) building level. Based on the results produced from an electricity database of multi-family residential buildings in New York City (NYC), the CLSAF model can achieve higher prediction accuracy compared to 4 classical benchmark models. The CLSAF model enables durable robustness by leveraging the advantages of its autocorrelation-based feature-selection algorithm and a model-simplification method was developed to prevent overfitting when confronted with volatile load data caused by changes in unpredictable resident behaviors.

With the same electricity database from multi-family residential buildings considered in Chapter 2, a further analysis was conducted in Chapter 3 to identify and predict the growth in

residential electricity usage in New York City associated with the Covid-19 impact. Special contribution was given to two characteristics of residential electricity usage: 1) the electricity use (kWh) of an average apartment on weekdays from 9am to 5pm, and 2) the hourly peak demand (Watt) of an average apartment between 12pm and 5pm. Two important factors were identified and considered as the essential predictors for forecasting the two characteristics, which are namely, the severity of the pandemic and the outdoor Wet Bulb Temperature (WBT). A series of regression models were built upon these two factors, which can predict the two characteristics with an R^2 of 0.56-0.57 for days when no cooling is required and 0.80-0.84 for warmer days. By performing Monte Carlo simulations, these regression models can be used to forecast the two usage characteristics for conditions which, fortunately, did not actually occur in 2020, but which could occur in the future in NYC, in similar regions, or indeed in future pandemics or natural catastrophes with comparable stay-at-home guidelines. Under such assumed future conditions, the weekday 8-hour-electricity-use (9am-5pm) could be 15%—24% higher than the one under normal circumstances. The weekday 5-hour-peak-demand (12pm-5pm) under the assumed condition could be 35%—53% higher than otherwise, where the highest point of the simulation results could be twice the maximum 5-hour peak demand in 2019 (894 watts).

Starting from Chapter 4, the focus of this dissertation turned to the newly developed data-driven methods in SHM. In Chapter 4, a New Generalized Autoencoder (NGAE) framework, integrated with a statistical-pattern-recognition-based approach that uses power cepstral coefficients as the Damage Sensitive Features (DSFs), was developed for the problem of structural damage detection and quantification in an unsupervised-learning manner. The NGAE can effectively characterize the overall structural properties embedded in the cepstral coefficients thanks to a newly defined encoder-decoder mapping, largely reducing the effects of the variance

attributed to the external excitation and to measurement noise. This mapping results in an appreciable accuracy in the assessment of damage within the structural system. The effectiveness of the NGAE has been validated through numerical as well as experimental data, namely, an 8 DOF shear-type system excited by an external force, and the benchmark problem of the Z24 bridge in Switzerland with various types of undamaged and damaged conditions.

Driven by the motivation to recognize various damage scenarios rather than just detecting the presence of damage as in Chapter 4, the problem of structural damage classification was deeply explored in Chapter 5. For this problem, a novel data augmentation strategy based on a Conditional Variational Autoencoder (CVAE) architecture was developed so to create a "balanced" dataset of the cepstral coefficients of the structural acceleration response. This augmented dataset can be used to systematically build a Probabilistic Linear Discriminant Analysis (PLDA) model for damage identification and classification. The proposed data augmentation strategy can effectively address the issue, commonly found in monitoring of real civil structures, of limited datasets from structures in damaged conditions. The PLDA model, trained with the augmented balanced dataset, can performs well for structural damage identification and classification in both supervised- and unsupervised-learning manners. The proposed data augmentation strategy and the structural damage classification method have been validated through two case studies, namely, an 8 DOF system model excited by different random Gaussian signals, and the same real-bridge structure (the Z24 bridge) considered in Chapter 4.

Finally, based on the results in Chapters 4 and 5, the problem of structural damage localization for both linear and nonlinear systems is studied in Chapter 6, where a novel data-driven modeling method is proposed. This method is based on the Linear Discriminant Analysis (LDA) and on the structural local characteristics embedded in the cepstral coefficients of structural acceleration

response. The developed LDA model is able to highlight the structural local characteristics of the cepstral coefficients in the LDA latent space thanks to its capability to maximize the separation distance between clusters. With a proper metric as a damage index to quantify the damage levels of individual recording locations of a monitored system, the localization of damaged areas within the structure can be accurately achieved by comparing the distributions of the projected cepstral coefficients in the LDA latent space between undamaged and damaged conditions. The effectiveness of the proposed method has been verified through two case studies, i.e., an 8 DOF linear system and a 4 DOF nonlinear system.

In terms of the form of data used, this dissertation is focused on the acquisition, analysis, and modeling of recorded time-series data for different problem objectives. The main "common" operations include data wrangling and cleaning, feature extraction and selection, and algorithm development and validations. For the two REM problems covered in Chapters 2 and 3, i.e., the short-term electricity load forecasting and peak load demand forecasting, they can be considered as typical scalar-based time series forecasting problems. The key to better solving these problems is to improve the models so to be as accurate as possible to achieve excellent regression performance. This can be done through continuous improvement of data quality and modeling strategies, such as extensive statistical analysis to identify important factors (predictors), extracting the key features from raw data to enhance relevant information, and modifying model frameworks to better meet specific requirements of the problems (e.g., the CLSAF model developed in Chapter 2 to address the problem of high volatility load data). For the SHM problems studied in Chapters 4-6, the main objective of developing data-driven models is to achieve a correct description of the monitored structural system, i.e., to build a surrogate data-based model that represents the monitored structural system in its undamaged operational state and that is capable to highlight

anomalies when damage occurs. Therefore, the models developed for these problems must not only produce appreciable regression/fitting accuracies with respect to measured response data or related DSFs (e.g., the cepstral coefficients extracted and used in Chapters 4 – 6), but also provide a reasonable characterization and ideal generalization of the structural physical properties of the monitored system.

7.2 Future directions

Based on the research progress achieved so far, the following research directions for the two areas of REM and SHM are recommended for future investigation.

1) For the REM area, it is suggested to explore in depth the development of higher-level functional modules for electricity load control and forecasting based on the previously developed methods and intuitions. The main motivation is that, although the electricity load modeling techniques (for both the short-term and the long-term forecasting problems) have been extensively explored by scholars over past decades, there is still much room for improvement in the electricity modeling of multi-step ahead forecasting strategies. It should be noted that this multi-step ahead forecasting is quite different from a single-step forecasting for an hour, day, or year into the future; it is a simultaneous forecasting of multiple time steps for hours, days, or years into the future. Generally, such a multi-step ahead forecasting objective is extremely challenging because of relatively smaller spatial granularities, especially for the individual apartments, as discussed in Chapters 2 – 3. This is because a sophisticate mathematical or machine learning model can easily overfit electricity data measured in individual households due to the associated idiosyncratic human behaviors (an apartment's temporary vacancy, for example).

A tentatively conceived, potentially effective solution for the multi-step ahead forecasting problem is that of employing the attention mechanism on top of the existing powerful deep learning

frameworks, e.g., incorporating the transformer self-attention mechanism [39] into a general Multi-Layer Perceptron (MLP) framework to perform the electricity modeling. The main advantage for using the attention mechanism is that one can maximize the extraction of the underlying load profiles from historical electricity load measurements in a more rational and intelligent way, thanks to its function of optimizing weight allocation to the key autocorrelation information of input sequence data. Consequently, not only can the information of residential load profiles be used more efficiently to achieve the primary objective of multi-step ahead load forecasting, but a smarter data-driven modeling method, compared to the previously proposed method (Chapter 2) that requires to be integrated with an additional optimal feature selection algorithm, can be obtained.

This multi-step ahead electricity load forecasting method can play a crucial role in electricity load peak control and demand response management, which could serve as a rather useful tool for electric utility companies. This is because such a technology can be powerfully integrated with various applications of cutting-edge smart grids/batteries, helping to more accurately indicate the occurrence of peak demand and providing new strategies for optimizing electricity allocation. These possible advances in smart grid applications can indeed help address the common global challenge of daily unbalanced load distribution between peaks and troughs in the residential electricity sector, as discussed in Chapters 2-3.

2) For the SHM area, it is proposed, in the immediate future, to refine the validation process of the proposed structural damage localization method (as discussed in Section 6.4) by testing the method with more extensive datasets collected from real structures (e.g., bridges or buildings). The Z-24 dataset, although still widely used, is old and quite limited. In addition, it would be interesting

to test the proposed method in more complex damage scenarios, such as the cases of multiple damage locations.

It is also proposed to develop machine learning models that are capable to directly account for some physical properties of structures. Over the past few years, there have been many studies focused on the development of the Physics-Informed Neural Networks (PINNs) [4, 116] in the fields of science and engineering, e.g., re-configuring the structures of MLPs or Recurrent Neural Networks (RNNs), and/or customizing the associated loss functions, to fulfill the physical laws of the governing differential equations followed by observed data.

A promising research direction for addressing the vibration-based SHM problems through the PINNs is to modify or upgrade the framework of Neural Ordinary Differential Equations (Neural ODEs) [117], which is an important member in the PINN community. The key intuition behind the original Neural ODEs is to learn the underlying governing dynamics (differential equations) from the measurement data of dynamical systems via the mechanism of hidden-layer-output transition of the Residual Neural Network (ResNet) [118]. Such a Neural ODEs framework provides a new paradigm and insights into the linkage of neural networks with differential equations. Hence, by developing a data-driven model on the basis of a Neural ODEs framework, the governing vibration differential equation embedded in the measured response data of a monitored structural system can be well described.

An alternative strategy of the PINN modeling for the vibration-based SHM problems is to customize the loss functions of classical neural networks to discover the governing dynamics or to identify key structural parameters. The basic idea is to add the necessary "physical terms" as additional losses to the original loss function consisting of only regression/fitting errors (e.g., the mean squared error). These terms are added to satisfy the laws of the governing differential

equations and the associated boundary conditions (if known), with/without the unknown structural parameters to be identified (e.g., the natural frequencies). In this way, this PINN model can be trained with the measured structural response data for a system identification objective to get the key structural parameters of the monitored system, and the trained model can subsequently become a data-based surrogate model for that system. This built surrogate PINN model, for example, can be then incorporated into a model updating framework for various structural damage assessment problems.

References

[1] Lu P, Chen S, Zheng Y. Artificial intelligence in civil engineering. *Mathematical Problems in Engineering*, vol. 2012, 2012.

[2] Pregnolato M, Gunner S, Voyagaki E, De Risi R, Carhart N, Gavriel G, et al. Towards Civil Engineering 4.0: Concept, workflow and application of Digital Twins for existing infrastructure. *Automation in Construction*, 141:104421, 2022.

[3] Lagaros ND, Plevris V. Artificial intelligence (AI) applied in civil engineering. *Applied Sciences*, 12(15):7595, 2022.

[4] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422-40, 2021.

[5] LeCun Y. 1.1 deep learning hardware: past, present, and future. In *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 12-19, 2019.

[6] Azimi M, Eslamlou AD, Pekcan G. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. *Sensors*, 20(10):2778, 2020.

[7] Bourdeau M, qiang Zhai X, Nefzaoui E, Guo X, Chatellier P. Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustainable Cities and Society*, 48:101533, 2019.

[8] Yildiz B, Bilbao JI, Dore J, Sproul AB. Recent advances in the analysis of residential electricity consumption and applications of smart meter data. *Applied Energy*, 208:402-27, 2017.

[9] Edwards RE, New J, Parker LE. Predicting future hourly residential electrical consumption: A machine learning case study. *Energy and Buildings*, 49:591-603, 2012.

[10] Siano P. Demand response and smart grids—A survey. *Renewable and sustainable energy reviews*, 30:461-78, 2014.

[11] An Y, Chatzi E, Sim SH, Laflamme S, Blachowski B, Ou J. Recent progress and future trends on damage identification methods for bridge structures. *Structural Control and Health Monitoring*, 26(10):e2416, 2019.

[12] Mashayekhi M, Santini‐Bell E. Three-dimensional multiscale finite element models for in-service performance assessment of bridges. *Computer-Aided Civil and Infrastructure Engineering*. 34(5):385-401, 2019.

[13] Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*. 115:213-37, 2019.

[14] Gross G, Galiana FD. Short-term load forecasting. *Proceedings of the IEEE*, 75(12):1558-73, 1987.

[15] Li L, Meinrenken CJ, Modi V, Culligan PJ. Short-term apartment-level load forecasting using a modified neural network with selected auto-regressive features. *Applied Energy*, 287:116509, 2021.

[16] Wang N. Transactive control for connected homes and neighbourhoods. *Nature Energy*, 3(11):907-9, 2018.

[17] Zheng M, Meinrenken CJ, Lackner KS. Smart households: Dispatch strategies and economic analysis of distributed energy storage for residential peak shaving. *Applied Energy*, 147:246-57, 2015.

[18] Song Y, Ding Y, Siano P, Meinrenken C, Zheng M, Strbac G. Optimization methods and advanced applications for smart energy systems considering grid-interactive demand response, 2020.

[19] Javed F, Arshad N, Wallin F, Vassileva I, Dahlquist E. Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting. *Applied Energy*, 96:150-60, 2012.

[20] Meinrenken CJ, Mehmani A. Concurrent optimization of thermal and electric storage in commercial buildings to reduce operating cost and demand peaks under time-of-use tariffs. *Applied Energy*, 254:113630, 2019.

[21] Amasyali K, El-Gohary NM. A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews*, 81:1192-205, 2018.

[22] Ghofrani M, Hassanzadeh M, Etezadi-Amoli M, Fadali MS. Smart meter based short-term load forecasting for residential customers. In *2011 North American Power Symposium*, pages 1-5, 2011.

[23] Munkhammar J, van der Meer D, Widén J. Very short term load forecasting of residential electricity consumption using the Markov-chain mixture distribution (MCM) model. *Applied Energy*, 282:116180, 2021.

[24] Zheng J, Xu C, Zhang Z, Li X. Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. In *2017 51st Annual conference on information sciences and systems (CISS)*, pages 1-6, 2017.

[25] Marino DL, Amarasinghe K, Manic M. Building energy load forecasting using deep neural networks. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 7046-7051, 2016.

[26] Andriopoulos N, Magklaras A, Birbas A, Papalexopoulos A, Valouxis C, Daskalaki S, et al. Short term electric load forecasting based on data transformation and statistical machine learning. *Applied Sciences*, 11(1):158, 2020.

[27] Jain RK, Smith KM, Culligan PJ, Taylor JE. Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy. *Applied Energy*, 123:168-78, 2014.

[28] Zheng Z, Chen H, Luo X. A Kalman filter-based bottom-up approach for household short-term load forecast. *Applied Energy*, 250:882-94, 2019.

[29] Xu L, Wang S, Tang R. Probabilistic load forecasting for buildings considering weather forecasting uncertainty and uncertain peak load. *Applied energy*, 237:180-95, 2019.

[30] Chen H, Wang S, Wang S, Li Y. Day-ahead aggregated load forecasting based on two-terminal sparse coding and deep neural network fusion. *Electric Power Systems Research*, 177:105987, 2019.

[31] Jain RK, Damoulas T, Kontokosta CE. Towards data-driven energy consumption forecasting of multi-family residential buildings: feature selection via the lasso. In *Computing in Civil and Building Engineering*, pages 1675-1682, 2014.

[32] Candanedo LM, Feldheim V, Deramaix D. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and buildings*, 140:81-97, 2017.

[33] Amarasinghe K, Marino DL, Manic M. Deep neural networks for energy load forecasting. In *2017 IEEE 26th international symposium on industrial electronics (ISIE)*, pages 1483-1488, 2017.

[34] Wang S, Wang X, Wang S, Wang D. Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting. *International Journal of Electrical Power & Energy Systems*, 109:470-9, 2019.

[35] Wan R, Mei S, Wang J, Liu M, Yang F. Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting. *Electronics*, 8(8):876, 2019.

[36] Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, vol. 28, 2015.

[37] Meinrenken CJ, Rauschkolb N, Abrol S, Chakrabarty T, Decalf VC, Hidey C, et al. MFRED, 10 second interval real and reactive power for groups of 390 US apartments of varying size and vintage. *Scientific Data*. 7(1):375, 2020.

[38] Hyndman RJ. Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, 4(4):43-6, 2006.

[39] Hewamalage H, Bergmeir C, Bandara K. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388-427, 2021.

[40] Hyndman RJ, Khandakar Y. Automatic time series forecasting: the forecast package for R. *Journal of statistical software*, 27:1-22, 2008.

[41] Spyers-Ashby JM, Bain PG, Roberts SJ. A comparison of fast Fourier transform (FFT) and autoregressive (AR) spectral estimation techniques for the analysis of tremor data. *Journal of neuroscience methods*, 83(1):35-43, 1998.

[42] Li L, Meinrenken CJ, Modi V, Culligan PJ. Impacts of COVID-19 related stay-at-home restrictions on residential electricity use and implications for future grid stability. *Energy and Buildings*, 251:111330, 2021.

[43] Abu-Rayash A, Dincer I. Analysis of the electricity demand trends amidst the COVID-19 coronavirus pandemic. *Energy Research & Social Science*, 68:101682, 2020.

[44] Chen CF, de Rubens GZ, Xu X, Li J. Coronavirus comes home? Energy use, home energy management, and the social-psychological factors of COVID-19. *Energy research & social science*, 68:101688, 2020.

[45] Bahmanyar A, Estebsari A, Ernst D. The impact of different COVID-19 containment measures on electricity consumption in Europe. *Energy Research & Social Science*. 68:101683, 2020.

[46] Edomah N, Ndulue G. Energy transition in a lockdown: An analysis of the impact of COVID-19 on changes in electricity demand in Lagos Nigeria. *Global Transitions*, 2:127-37, 2020.

[47] Wilson G, Godfrey N, Sharma S, Bassett T. We analysed electricity demand and found coronavirus has turned weekdays into weekends. *The Conversation*, 2020.

[48] Coronavirus BB. Domestic Electricity use up During Day as Nation Works From Home, 2020.

[49] Hoang AT, Nguyen XP, Le AT, Huynh TT, Pham VV. COVID-19 and the global shift progress to clean energy. *Journal of Energy Resources Technology*, 143(9), 2021.

[50] Liedtke M, Bussewitz C. Damage from virus: utility bills overwhelm some households. *Associate Press news*, 2021.

[51] Meinrenken CJ, Abrol S, Gite GB, Hidey C, McKeown K, Mehmani A, et al. Residential electricity conservation in response to auto-generated, multi-featured, personalized eco-feedback designed for large scale applications with utilities. *Energy and Buildings*, 232:110652, 2021.

[52] Fernandez NE, Katipamula S, Wang W, Xie Y, Zhao M, Corbin CD. *Impacts of commercial building controls on energy savings and peak load reduction*. Pacific Northwest National Lab. (PNNL), Richland, WA (United States), 2017.

[53] Daniel WW. Kolmogorov–Smirnov one-sample test. *Applied nonparametric statistics*, vol. 2, 1990.

[54] Li L, Morgantini M, Betti R. Structural damage assessment through a new generalized autoencoder with features in the quefrency domain. *Mechanical Systems and Signal Processing*, 184:109713, 2023.

[55] Morgantini M, Betti R, Balsamo L. Structural damage assessment through features in quefrency domain. *Mechanical Systems and Signal Processing*, 147:107017, 2021.

[56] Das S, Saha P, Patro SK. Vibration-based damage detection techniques used for health monitoring of structures: a review. *Journal of Civil Structural Health Monitoring*, 6:477-507, 2016.

[57] Shih HW, Thambiratnam DP, Chan TH. Damage detection in slab‐on‐girder bridges using vibration characteristics. *Structural Control and Health Monitoring*, 20(10):1271-90, 2013.

[58] Brownjohn JM, Xia PQ, Hao H, Xia Y. Civil structure condition assessment by FE model updating:: methodology and case studies. *Finite elements in analysis and design*, 37(10):761-75, 2001.

[59] Balsamo L, Betti R, Beigi H. A structural health monitoring strategy using cepstral features. *Journal of Sound and Vibration*, 333(19):4526-42, 2014.

[60] Bogert BP. The quefrency alanysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. In *Proc. Symposium Time Series Analysis*, pages 209-243, 1963.

[61] Zhang G, Harichandran RS, Ramuhalli P. Application of noise cancelling and damage detection algorithms in NDE of concrete bridge decks using impact signals. *Journal of Nondestructive Evaluation*, pages 259-72, 2011.

[62] Avci O, Abdeljaber O, Kiranyaz S, Hussein M, Gabbouj M, Inman DJ. 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, 2021.

[63] O'Shea K, Nash R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015

[64] Abdeljaber O, Avci O, Kiranyaz S, Gabbouj M, Inman DJ. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388:154-70, 2017.

[65] Cha YJ, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361-78, 2017.

[66] Wang Z, Cha YJ. Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage. *Structural Health Monitoring*, 20(1):406-25, 2021.

[67] Pathirage CS, Li J, Li L, Hao H, Liu W, Ni P. Structural damage identification based on autoencoder neural networks and deep learning. *Engineering structures*, 172:13-28, 2018.

[68] Ma X, Lin Y, Nie Z, Ma H. Structural damage identification based on unsupervised feature-extraction via Variational Auto-encoder. *Measurement*, 160:107811, 2020.

[69] Bejani MM, Ghatee M. A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*, pages 1-48, 2021.

[70] Kramer MA. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233-43, 1991.

[71] Wold S, Esbensen K, Geladi P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37-52, 1987.

[72] Tschannen M, Bachem O, Lucic M. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[73] Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[74] Bourlard H, Kamp Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4-5):291-4, 1988.

[75] Wang W, Huang Y, Wang Y, Wang L. Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 490-497, 2014.

[76] Da Silva IN, Hernane Spatti D, Andrade Flauzino R, Liboni LH, dos Reis Alves SF, da Silva IN, et al. *Artificial neural network architectures and training processes*. Springer International Publishing, 2017.

[77] Boger Z, Guterman H. Knowledge extraction from artificial neural network models. In *1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, 4:3030-3035, 1997.

[78] Goodfellow I, Bengio Y, Courville A. *Deep learning*, MIT press, 2016.

[79] McLachlan GJ. Mahalanobis distance. *Resonance*, 4(6):20-6, 1999.

[80] Ververidis D, Kotropoulos C. Information loss of the mahalanobis distance in high dimensions: Application to feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2275-81, 2009.

[81] Krämer C, De Smet CA, De Roeck G. Z24 bridge damage detection tests. In *IMAC 17, the International Modal Analysis Conference*, 3727:1023-1029, 1999.

[82] Reynders E, De Roeck G. *Vibration-based damage identification: the Z24 benchmark*, 2014.

[83] Dike HU, Zhou Y, Deveerasetty KK, Wu Q. Unsupervised learning based on artificial neural network: A review. In *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pages 322-327, 2018.

[84] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. *Journal of Big data*, 3(1):1-40, 2016.

[85] Tronci EM, Beigi H, Feng MQ, Betti R. Transfer Learning from Audio Domains a valuable tool for Structural Health Monitoring. In *Dynamics of Civil Structures, Volume 2: Proceedings of the 39th IMAC, A Conference and Exposition on Structural Dynamics 2021*, pages 99-107, 2022.

[86] Feng SY, Gangal V, Wei J, Chandar S, Vosoughi S, Mitamura T, et al. A survey of data augmentation approaches for NLP. *arXiv preprint arXiv:2105.03075*, 2021.

[87] Perez L, Wang J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[88] Zhai G, Narazaki Y, Wang S, Shajihan SA, Spencer Jr BF. Synthetic data augmentation for pixel-wise steel fatigue crack identification using fully convolutional networks. *Smart Struct Syst*, 29(1):237-50, 2022.

[89] Wan P, He H, Guo L, Yang J, Li J. InfoGAN-MSF: a data augmentation approach for correlative bridge monitoring factors. *Measurement Science and Technology*, 32(11):114008, 2021.

[90] Sohn K, Lee H, Yan X. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, vol. 28, 2015.

[91] Ioffe S. Probabilistic linear discriminant analysis. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV 9*, pages 531-542, 2006.

[92] Kingma DP, Welling M. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307-92, 2019.

[93] Jordan, MI. Graphical models, Statist. Sci, 19(1):140-155, 2004.

[94] Kingma DP, Welling M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[95] An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1-8, 2015.

[96] Izenman AJ, Izenman AJ. Linear discriminant analysis. *Modern multivariate statistical techniques: regression, classification, and manifold learning*, pages 237-80, 2008.

[97] Nandakumar K, Chen Y, Dass SC, Jain A. Likelihood ratio-based biometric score fusion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):342-7, 2007.

[98] Giordano PF, Limongelli MP. Response-based time-invariant methods for damage localization on a concrete bridge. *Structural Concrete*, 21(4):1254-71, 2020.

[99] Giordano PF, Prendergast LJ, Limongelli MP. A framework for assessing the value of information for health monitoring of scoured bridges. *Journal of Civil Structural Health Monitoring*, 10:485-96, 2020.

[100] Friswell M, Mottershead JE. *Finite element model updating in structural dynamics*. Springer Science & Business Media, 1995.

[101] Peeters B, De Roeck G. Stochastic system identification for operational modal analysis: a review. *J. Dyn. Sys., Meas., Control*, 123(4):659-67, 2001.

[102] Dems K, Mróz Z. Identification of damage in beam and plate structures using parameter-dependent frequency changes. *Engineering Computations*, 18(1/2):96-120, 2001.

[103] Pandey AK, Biswas M, Samman MM. Damage detection from changes in curvature mode shapes. *Journal of sound and vibration*, 145(2):321-32, 1991.

[104] Stubbs N, Kim JT, Topole K. An efficient and robust algorithm for damage localization in offshore platforms. *In Proceedings of the ASCE 10th structures congress*, 1:543-546, 1992.

[105] Limongelli MP. The interpolation damage detection method for frames under seismic excitation. *Journal of Sound and Vibration*, 330(22):5474-89, 2011.

[106] Jiang X, Ma ZJ, Ren WX. Crack detection from the slope of the mode shape using complex continuous wavelet transform. *Computer-Aided Civil and Infrastructure Engineering*, 27(3):187-201, 2012.

[107] Solís M, Algaba M, Galvín P. Continuous wavelet analysis of mode shapes differences for damage detection. *Mechanical Systems and Signal Processing*, 40(2):645-66, 2013.

[108] Ng CT, Veidt M. A Lamb-wave-based technique for damage detection in composite laminates. *Smart materials and structures*, 18(7):074006, 2009.

[109] Mitra M, Gopalakrishnan S. Guided wave based structural health monitoring: A review. *Smart Materials and Structures*, 25(5):053001, 2016.

[110] Zhang S, Li CM, Ye W. Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network. *Mechanical Systems and Signal Processing*, 147:107107, 2021.

[111] Pan VY. Solving a polynomial equation: some history and recent progress. *SIAM review*, 39(2):187-220, 1997.

[112] Fisher RA. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179-88, 1936.

[113] Bouc R. Forced vibrations of mechanical systems with hysteresis. In *Proc. of the Fourth Conference on Nonlinear Oscillations, Prague*, 1967.

[114] Wen YK. Methods of random vibration for inelastic structures, *Applied Mechanics Reviews*, 42(2):39-52, 1989.

[115] Lin JW, Betti R, Smyth AW, Longman RW. On-line identification of non-linear hysteretic structural systems using a variable trace approach. *Earthquake engineering & structural dynamics*, 30(9):1279-303, 2001.

[116] Lai Z, Mylonas C, Nagarajaiah S, Chatzi E. Structural identification with physics-informed neural ordinary differential equations. *Journal of Sound and Vibration*, 508:116196, 2021.

[117] Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. *Advances in neural information processing systems*, vol. 31, 2018.

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