UC Berkeley
UC Berkeley Electronic Theses and Dissertations

Title
Data-Centric Machine Learning for Human-Centric Applications

Permalink
https://escholarship.org/uc/item/4mq363gb

Author
Das, Hari Prasanna

Publication Date
2023

Peer reviewed|Thesis/dissertation

Data-Centric Machine Learning for Human-Centric Applications

By

Hari Prasanna Das

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Costas J. Spanos, Chair
Professor Alberto Sangiovanni-Vincentelli
Professor Stefano Schiavon

Summer 2023

Data-Centric Machine Learning for Human-Centric Applications

Abstract

Data-Centric Machine Learning for Human-Centric Applications

by

Hari Prasanna Das

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Costas J. Spanos, Chair

Climate change and pandemics are two of the most pressing threats facing humanity today. Addressing these urgent threats require immediate mitigative actions. In the US, buildings are responsible for 40% of primary energy consumption, 73% of electrical use and 40% of greenhouse gas emissions, the primary cause of global warming, and such high levels are now rapidly spreading across the rest of the world. At the same time, buildings are integral to human lives, as we spend most of our time in them which substantially affects our health and productivity. So, for climate change mitigation, it is essential to optimize energy use in buildings while ensuring human comfort. On the other hand, for pandemics mitigation, it is crucial to diagnose and have a better understanding of the new disease in a time-sensitive manner. Over the years, Machine Learning (ML) as a tool has been widely utilized for both the above efforts. However, both buildings and pandemic-specific healthcare systems exhibit a number of shared data-specific challenges, hindering robust ML implementations.

We will present 3 major research works on tackling them with generative modeling, and transfer learning. The first work will be on conditional synthetic data generation, where the focus is to conditionally generate synthetic data for classes with infrequent data points. The applications include tackling class imbalance in healthcare data, and privacy-preserving data sharing. The second will be on improved pre-processing methods for tabular data (a common data type in smart buildings) to enable seamless use by many ML algorithms. To improve the generalizability and scalability of the models, the third work will be on a transfer learning-based adversarial domain adaptation method, with applications in adapting personal thermal comfort models in buildings from one occupant to another without using any data labels for the target occupant. With this method, the time and the resource-intensive task of acquiring multiple labels for the target environment in a building can be avoided.

Dedicated to my parents,
Muralidhar Das, and Jayanti Gachhayat

Contents

II Conditional Synthetic Data Generation

5 Conditional Synthetic Data Generation

III Tackling Data and Model Inconsistencies

6 Improved Tabular Data Pre-Processing Methods

IV Transfer Learning: Cross-Domain Prediction and Generation

7 Cross-Domain Classifier Adaptation

8 Cross-Domain Conditional Synthetic Generation

List of Figures

List of Tables

Acknowledgments

I express my heartfelt gratitude to my advisor Prof. Costas J. Spanos. I joined your group as a Power Systems researcher, with topical awareness of the fields of my current research: buildings, and machine learning. From then to now, your guidance, friendship, support, and advice have shaped me into what I am now. Thank you for believing in me and constantly encouraging me to explore the domains and methods I was passionate about. I am always amazed at how much down to earth, calm, and friendly you are, while wearing so many leadership hats. Whenever I faced an issue, even outside of my Ph.D., you always listened to and helped me with it. You always ensured we have desired situations and environments to focus on research. I feel fortunate and lucky to have you as my advisor, and I wish to imbibe your qualities in my life.

I learnt so much about thinking properly, and polishing research ideas from Prof. Alberto Sangiovanni-Vincentelli. Thank you for guiding me in writing the two successful C3 DTI grant proposals. It gave me hands-on experience on how to present futuristic ideas, while grounding on relevant experience and existing research. I would like to thank Prof. Stefano Schiavon, who has played such a major part in my Ph.D. I remember meeting you just after starting my Ph.D. and discussing the personal thermal comfort research, and being amazed at how knowledgeable and friendly you are. Over the years, you have mentored me not just in research, but also on how to develop myself professionally. I hope to get your advise throughout all future endeavors in my life. Both, Alberto and Stefano were in my dissertation committee, thank you for shaping my research over the years.

It was in Prof. Pieter Abbeel's Deep Unsupervised Learning class that I learned about generative modeling, which I later based my research on. Thank you Pieter, for imbibing above knowledge in me, guiding me in research works, being in my qualification exam committee, and overall being a great mentor. Prof. Alex Bayen and Prof. Venkat Anantharam were kind enough to have me in their teaching teams for EECS 127/227AT: Optimization Models in Engineering, and provided me with great responsibilities of leading the course contents team. Thank you Alex and Venkat for believing in me, and teaching me how to teach.

Prof. Ashok Pradhan, my undergraduate research mentor has a major role in where I am today. He motivated me to pursue research and Ph.D., while mentoring me for my B.Tech project. It was with him that I published my first paper. He has always treated me like his family, I cannot thank him enough for all he has done for me.

I also would like to thank Prof. Zoltan Nagy, Dr. Draguna Vrabie, and Dr. Tianzhen Hong for being great mentors for me.

Thank you to Ioannis Konstantakopoulos and Abhinav Sethy for hosting me at Amazon Alexa for a summer internship in 2021. Ioannis, you have been my mentor since my first few days at Berkeley. We have discussed so much about research, career, and life together, and you have always guided me. Thank you for everything. Thank you to Wei Xu, and Lin Ma for a rewardable learning experience at ByteDance during my summer internship in 2022.

To all my research collaborators and friends over the years, Ming Jin, Ruoxi Jia, Han Zou, Ioannis Konstantakopoulos, Shichao Liu, Ryan Tran, Japjot Singh, Geoff Tison, Clayton

Chapter 1

Introduction

Figure 1.1: A taxonomy of the Smart Buildings illustrated at three levels: *cluster of buildings*, *single building*, and *occupant*.

Climate change and pandemics are two of the most pressing threats facing humanity today. Addressing these urgent threats requires immediate mitigative actions. In the US, buildings are responsible for 40% of primary energy consumption, 73% of electrical use and 40% of greenhouse gas emissions [2], the primary cause of global warming, and such high levels are now rapidly spreading across the rest of the world. At the same time, we spend 90% of our time every day in indoor environments, so buildings substantially influence our health, well-being, safety, and work and study performance. So, for climate change mitigation, it is essential to optimize energy use in buildings while ensuring human comfort. On the other hand, for pandemic mitigation, it is crucial to diagnose and have a better understanding of the new disease in a time-sensitive manner. Over the years, Machine Learning (ML) as a tool has been widely utilized for both the above efforts. However, both buildings and

pandemic-specific healthcare systems exhibit a number of shared data-specific challenges, hindering robust ML implementations.

In the following sections, we will cover more about the machine learning applications in smart buildings and pandemic specific healthcare, and point some data specific challenges present in them. We will then present our research work to tackle the above challenges.

1.1 ML Applications and Data Challenges in Smart Buildings

It is of utmost importance to improve building energy systems to optimize energy usage and thus limit the greenhouse gas emissions contributed by them, while, at the same time, ensuring an occupant-friendly environment to improve well-being and productivity. Energy use reductions in buildings can be an environmentally sustainable, equitable, cost-effective, and scalable approach to reducing greenhouse gas emissions. Simultaneously, maintaining occupant comfort and productivity is crucial in achieving occupant satisfaction in buildings.

The smart building ecosystem is illustrated in Fig 1.1. Occupants constitute the basic building block of the ecosystem. Being the consumer of the enviornment that a building provides, occupants necessitate regulation of the building systems to achieve the desired environment. The building comprises of structures, devices and systems in place to control and maintain the desired environment for the occupants, along with diagnostic systems to ensure a robust operation. The building operation requires energy, which primarily comes in the form of electricity. The electrical energy is supplied to buildings via a power distribution system, where, with the advent of smart grids, buildings interact and exchange surplus energy and other ancillary services with the energy provider and with each other.

In efforts to improve energy efficiency in buildings, researchers and industry leaders have attempted to implement control and automation approaches alongside techniques like incentive design and price adjustment to more effectively regulate the energy usage [3]. The heterogeneity of user preferences in regard to building utilities is considerable in variety and necessitates a system that can adequately account for differences from one occupant to another. Focus has shifted towards modeling occupant behavior to incorporate their preferences in building control and automation [4]. The behavioral models can then be studied to introduce initiatives to encourage energy efficient behaviors among the occupants/energy users. With the growth of internet-of-things (IoT) devices, and the great variety of user-to-device and device-to-device interactions, there is a need for integration and coordination of the related objectives and actions. Further, to derive insights from the vast amount of data in certain scenarios, and from the limited amount of data in others, ML applications are proliferating in smart buildings. These ML-driven insights can be used for downstream tasks such as forecasting, prediction, and control.

Application of Machine Learning in Smart Buildings

	Inter-Building	Building	Occupant
Control strategies • Reinforcement learning • Hard control and model-predictive control	• Power flow coordination • Smart energy storage and distribution	• HVAC, lighting control • Space scheduling • Demand response	• Occupant-centric environmental control
Learning Models • Supervised learning • Physics-based learning • Generative modeling	• Demand response prediction • Building load prediction	• Building control • Fault diagnosis • Occupancy detection	• Thermal comfort • Activity detection • Utility Function Learning
Markets • Automated pricing • Time-series forecasting • Gamification	• Aggregation & mechanism design • Energy allocation	• Demand response cost optimization • Smart load scheduling	• Mechanism design based on occupant utility

Figure 1.2: Illustration of various applications where machine learning methods can be deployed in smart buildings, grouped at the cluster of buildings-level, the building-level, and the occupant-level.

Recent years have witnessed an exponential growth in machine learning implementation in smart buildings. Fig 1.2 illustrates the building components at various levels and classes of machine learning algorithms that are proposed for those systems.

Fault detection plays a crucial part in reducing maintenance costs and increasing the energy efficiency of building operations [5]. However, faults in an actual building don't occur frequently, and it is hard to collect fault data for analysis. Data sets for fault diagnosis are usually created through testbed experiments or simulations. Techniques in ML such as out-of-distribution detection has been proposed to detect faults [6].

ML has been utilized to predict thermal comfort. Often, the Fanger's features [7] (air temperature, mean radiant temperature, relative humidity, air velocity, clothing insulation and metabolic rate.), alone or with additional relevant real and synthetic features, are fed into a data-driven model to learn the connections between the features and the thermal preference labels [8,9]. The model is later leveraged to predict the same given raw features. Because the models are trained to learn directly from the data, and not from a rule established using prior experiments, they perform better as compared to the conventional Predicted Mean Vote (PMV) model, that functions by predicting a mean consensus of occupant comfort preference.

Kernel-based approaches have been widely-used for thermal sensation/preference prediction. The list of kernel-based methods popular for thermal comfort prediction includes

Support Vector Machine (SVM) [10–14], *K-Nearest Neighbors* (KNN) [8,15–18], and *Ensemble Learning* algorithms, such as *Random Forest* (RF) [8,9] and *AdaBoost* (Ab). Recently, *feed-forward neural networks* [17,19,20], and *time-series based networks* [21] have surpassed state-of-the-art kernel-based models in thermal comfort prediction.

Traditional thermal comfort models, as described above, are developed based on aggregated data from a large population. So, rather than predicting the thermal comfort of individuals, they were designed to predict the average thermal comfort of a population, when all its members are exposed to the same environment. This naturally misses the inevitable and sometimes significant differences in how different individuals respond to the same thermal environment. A new approach that uses personal comfort models instead of the average response of a large population can be applied to any building thermal control system [9]. Personal thermal comfort can adapt to the available input variables, such as environmental variables [22], occupant behaviors [23] and physiological signals [8]. ML algorithms ranging from kernel-based to neural network based methods have been proposed [9,24]. Other ML approaches are also becoming popular in modeling the complex interactions that exist between the features without much feature engineering, e.g. time-series prediction [25], artificial neural networks [19], etc. Better approaches for modeling tabular data in smart buildings, with a focus on thermal comfort datasets are provided in [26].

ML has also been proposed to be used in models that predict building performance. It has been used during the design stage to augment generative design and parametric simulations. Deep generative algorithms such as Generative Adversarial Networks (GANs) [27,28] have been proposed for generating diverse but realistic architectural floorplans, a process that has been known to be time-consuming iterative. The automated generation of architectural floorplans can be coupled with BPS tools to systematically explore architectural layouts that optimize building energy efficiency [29].

Metamodeling, defined as the practice of using a model to describe another model as an instance [30], is another aspect where machine learning has been extensively applied to BPS throughout the building lifecycle. Given the complex interaction between different building systems and sub-systems, design optimization during early design typically requires exploring a high-dimensional decision space. Consequently, machine learning has been used to create metamodels that can be used for optimization and uncertainty analysis [31,32].

Privacy is a crucial element in buildings, as it is linked to safety of occupants inhabiting them. Privacy preserving algorithms have been proposed to ensure machine learning algorithms designed for smart buildings do not compromise private information of occupants. Works such as [33] and [34] present accountable machine learning methods aimed to preserve privacy in cyber-physical systems such as buildings.

Occupancy and activity sensing are key aspects for the observability of a human-in-the-loop building control system. Traditionally, building operation methods that include occupancy as one of their parameters, such as starting heating/cooling from early morning till late in the evening during weekdays assuming maximum occupancy during working hours often have static schedules set for the occupancy, which is far from realistic. Also, how much a building will be occupied depends on several other factors, such as weather, building type, and

holiday schedule. Such static policies may lead to a significant waste in energy consumption, because the heating/cooling and ventilation levels are set with no regard for the actual occupancy level. Activity sensing also helps to provide personalized, context-aware services in buildings, thus enhancing overall satisfaction while creating a safety net for adverse events such as falls in elderly homes [35]. Occupancy sensing can be performed using both intrusive and non-intrusive methods. Intrusive methods require the occupants to carry an electronic device whose signature is followed by a central server to infer occupancy/positioning [36–40]. However, requiring occupants to constantly carry a device is not reliable. This problem gets magnified for the case of elderly population. Hence, non-intrusive methods for occupancy sensing are getting popular.

ML has been proposed for occupancy/activity sensing using data from modalities such as video and WiFi activity level. [41] use an U-Net like convolutional neural network on thermal images to infer occupancy. Other works employing similar machine learning methods on depth cameras are [42–44]. But, in general, cameras have other issues such as poor illumination conditions and occlusion. A recent body of work focuses on occupancy and activity detection from WiFi signals [45], because of their ubiquitous presence, and better privacy guarantees. Authors in [46, 47] use Channel State Information (CSI) data collected from WiFi sensors (a transmitter and a receiver) and measure the shape similarity between adjacent time series CSI curves to infer occupancy. Additional work improves the detection mechanism by using convolutional neural networks on the CSI heatmaps to detect human gestures [48]. Another modality that is used to detect occupancy is CO_2 data in a room. A number of works [49–52] employ machine learning methods to map the CO_2 concentration and occupancy. Finally, others propose sensor fusion, where data from multiple sensing modalities, i.e. RGB camera, and WiFi are used in tandem to come up with a robust activity detection mechanism [53].

Data-Specific Challenges

At the core of machine learning is data: its continuous availability, intelligent processing, efficient handling and storage. Smart buildings are equipped with an array of Internet-of-Things (IoT) devices that ensure the availability of rich data. The data is then fed to machine learning algorithms after appropriate curation and pre-processing to perform some task that achieves an objective, be it enhancing energy efficiency or improving occupant thermal comfort and productivity. For intelligent machine learning model design, it is crucial that the continuous availability of rich and diverse data from building systems is ensured. There are several data-specific challenges observed in smart buildings, as illustrated in Fig. 1.3.

A common challenge in designing ML based predictors in smart buildings is the issue of class-imbalance in data. For instance, almost all of the thermal comfort datasets [8, 54] are inherently data class-imbalanced, i.e. they have more data belonging to "Prefer No-Change"than "Prefer Warmer"and "Prefer Cooler"thermal preference classes. Researchers have tackled this issue by using weighted loss functions for ML models. A related challenge is the overall lack of sufficient amount of data for rare cases, such as aged and frail subjects [8] in
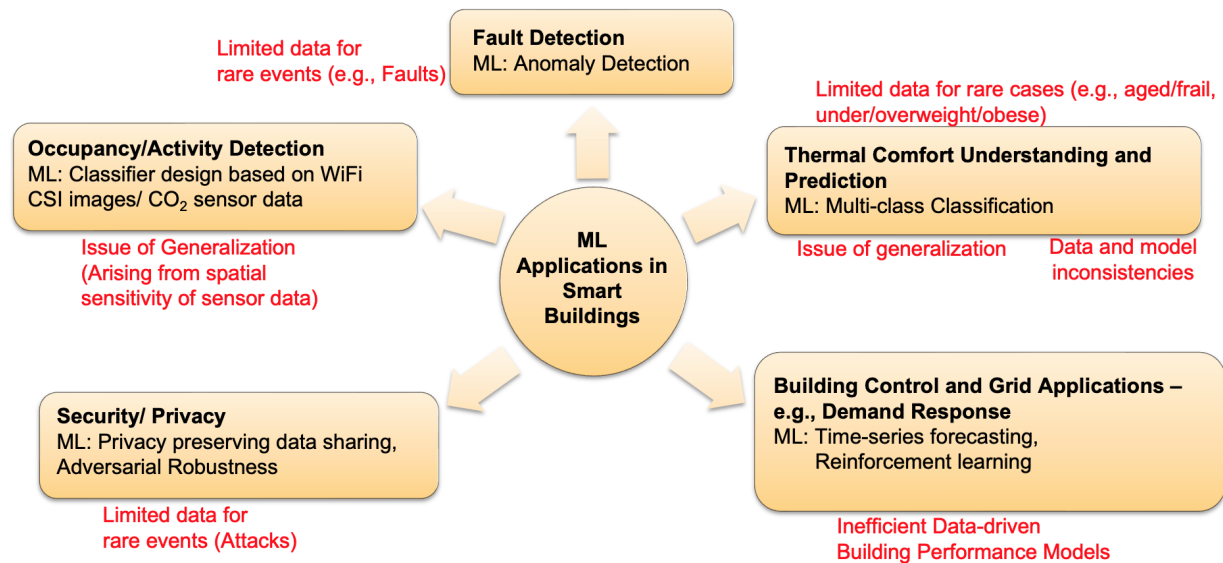
Figure 1.3: Machine Learning Applications and Data-related Challenges (in red font) in Smart Buildings

thermal comfort analysis, and fault scenarios in building fault detection. Collecting large amounts of data as required by ML models from humans and building systems via real-world experiments is expensive and cumbersome.

Another challenge is domain discrepancy. For instance, thermal comfort, as per Predicted Mean Vote (PMV) model (the widely adopted model for analyzing thermal comfort), is dependent upon 6 major parameters as described in the beginning of the section or, as per the adaptive thermal comfort model, is dependent on the outside temperature. However, it also varies from person to person, across climatic regions and economic conditions. A literature review of personal comfort models concluded that there is a lack of diversity in terms of building types, climates zone and participants that are considered in existing thermal comfort studies [55]. Under such domain discrepancy, models developed in one environment, when used in another target environment may lead to low accuracy or misleading predictions. Also, thermal comfort modeling depends largely on self-reporting, which is inherently unreliable.

Another data-specific challenge that exists in smart buildings is the compatibility between the available data, and state-of-the-art ML models. A large number of smart building datasets are tabular in nature. Tabular data is defined as data that is structured into rows, and columns of information. Each row contains the same number of cells (although some of these cells may be empty), which is considered as a single data sample. Each column in tabular data represents a variable, or a property or a feature of the system to which the dataset corresponds to. The columns in tabular datasets can be continuous, with variables whose values are real numbers, and can be uncountably infinite, or discrete, with variables that are categorical and can have a countably limited number of values. Continuous and
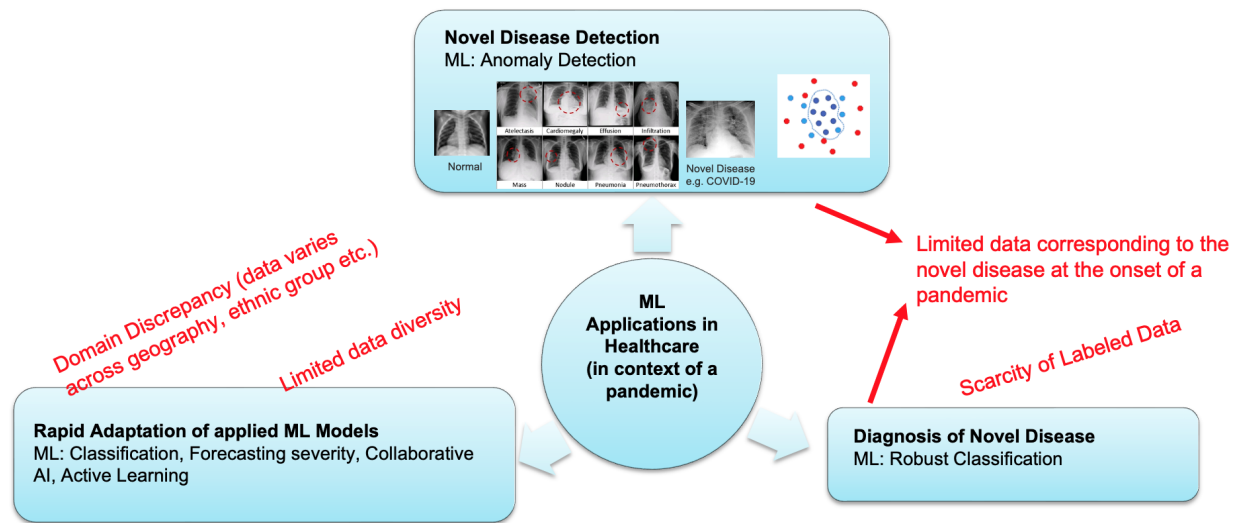
Figure 1.4: Machine Learning Applications and Data-related Challenges (in red font) in Pandemic Specific Healthcare

discrete features require specialized handling. Continuous variables generally observed in the real-world are multi-modal in nature, but existing practices to transform them during data pre-processing, namely min-max or gaussian normalization treat them as unimodal, which leads to inefficient modeling of the variables. Also, many of the widely used state-of-the-art ML models such as neural networks are smooth function approximators, and cannot be directly used with discrete data.

1.2 ML Applications and Data Challenges in Pandemic Specific Healthcare

The COVID-19 pandemic has created a public health crisis and continues to have a devastating impact on lives and healthcare systems worldwide. In the fight against this pandemic, a number of algorithms involving state-of-the-art machine learning techniques have been proposed. Data-based approaches have been used in a number of important tasks such as detection, mitigation, transmission modeling, decisions on lockdown, reopening and related restrictions etc. For example, ML tools such as out-of-distribution detection have been proposed [56] to detect that there has been a new disease in the world. Computer vision-based detection of COVID-19 from chest computed tomography (CT) images has been proposed as a supportive screening tool for COVID-19 [57], along with the primary diagnostic test of transcription polymerase chain reaction (RT-PCR). This is beneficial since obtaining definitive RT-PCR test results may take a lot of time in critical situations. Reinforcement learning based methods were also proposed to optimize mitigation policies that minimize the economic impact without

overwhelming the hospital capacity [58].

The application of machine learning algorithms in healthcare depends upon ample availability of disease data along with their attributes/labels. At the beginning of a pandemic, data corresponding to the disease might be unavailable or sparse. Sparse data often have limited variation in several important factors relevant to disease detection such as age, underlying medical conditions etc. Class imbalance is another issue faced by machine learning algorithms when pandemic-disease related data is limited. For example, at the onset of COVID-19, the amount of CT scan images corresponding to COVID-19 were much less than those corresponding to other existing lung diseases (e.g. pneumonia). ML models fed with such class-imbalanced data could be biased and thus provide inaccurate results. Furthermore, the amount of data with proper labels among the available pandemic data might be minimal. This issue can arise because healthcare professionals and domain experts who can review and label the data are busy treating patients inflicted with the new disease, or also because of privacy concerns associated with medical data sharing. Adapting ML models to the new disease under such label scarce scenarios needs special design.

Concurrently, after a new disease has been discovered, the healthcare ML tools must rapidly adapt to the new disease in order to assist medical professionals diagnose and treat affected individuals as quickly as possible. Rapid actions are also expected in the design of policy interventions that are based on insights from pandemic data. Another issue in development of machine learning algorithms for emerging pandemics is privacy. Development of solutions to pandemics at the scale of COVID-19 requires collaborative research which in turn presses the need for open-sourced healthcare data. But, even if healthcare organizations wish to release relevant data, they are often restricted in the amount of data to be released due to legal, privacy and other concerns.

1.3 Research Contributions

It can be observed that smart buildings and healthcare exhibit some shared data-specific challenges, such as class-imbalance and limited data for rare-events, domain discrepancy, and data-model inconsistencies. In this research, we aim to tackle the above challenges using tools of deep generative modeling and un/semi-supervised learning.

More specifically, we aim to generate synthetic data to augment the training dataset as one of the approaches for tackling challenges of class imbalance and limited data for rare events. Synthetic data generation can be done using classical methods such as SMOTE [59, 60] or using advanced neural network-based generative models [60–63]. To deal with the data/label insufficiency challenge across domains, we propose domain adaptation methods to adapt ML models from one domain to another. We also design pre-processing methods specifically for handling tabular data and enabling their use with smooth function approximators such as neural networks. Our proposed methods are generic, since they can also be used for other applications or data types with minor modifications.

The rest of this thesis has been divided into three parts, part I describing conditional synthetic data generation and its application in healthcare and smart buildings, part II covering transfer learning/domain adaptation for synthetic data generation and prediction, and part III introducing data pre-processing methods for handling tabular data. Finally, we present some avenues of future research.

Part I

Experimental Setup, Baseline Models, Problem Identification

Chapter 2

Personal Thermal Comfort Modeling

2.1 Introduction

Occupants' thermal comfort is associated with health [64,65], work productivity [66], learning performance [67] and well-being [68]. Indoor thermal environment design and thermostat settings in most buildings with mechanical systems rely on air temperature control values based on the existing predicted mean vote (PMV) model as described in thermal comfort standards as ASHRAE Standard 55 [69], EN 15251 [70] and ISO 7730 [71], while the adaptive model is used for automated buildings [72].

Nevertheless, neither PMV nor the adaptive model incorporates individual differences and dynamics in thermal perceptions. Also, both models ignore aspects of human thermoregulation and important personal psychophysics influencing the perception of thermal comfort [73]. The PMV predicts thermal sensation correctly only one out of three times and has a mean absolute error of one unit on the thermal sensation scale [74]. The main limitation of both PMV and adaptive models is that these two models were developed based on aggregated data from a large population. They were designed to predict the average thermal comfort of the entire population rather than that of an individual. Consequently, their accuracy on predicting thermal comfort for a specific occupant is very low. Kim et al. [9] proposed a framework of personal comfort models that can predict an individual's thermal comfort responses by leveraging Internet of Things (IoT) and machine learning, rather than the responses of an "average person." Such a framework has been applied in a few recent studies that aimed to customize thermal comfort models for each occupant through users' feedback, IoT and machine learning [75,76]. The primary advantage of a personal thermal comfort model lies in its capacity of self-learning and updating to suit an individual with a data-driven approach, resulting in higher predictive power.

Numerous recent studies have developed personal thermal comfort models by feeding different variables into machine learning algorithms. The three primary categories of variables are 1) environmental information, 2) occupant behavior, and 3) physiological signals. Probability distributions of thermal comfort for each occupant were created over indoor temperature

for HVAC controls [77]. A similar data-driven method with indoor environment was applied to classify occupants' personal thermal comfort with temperature and humidity sensors [78]. The second option is to track occupants' behavior to infer thermal comfort and preference, such as adjusting thermostats [79] or changing the settings of personal heating/cooling devices [80]. A personal comfort model using only control behavior of a smart chair system can generate a prediction AUC of 69% compared to approximately 53% (almost random) for the PMV and adaptive model [9]. Along with behavior-tracking, physiological signals, such as skin temperature [81], heart rate variability [82], electroencephalogram (EEG) [83], skin conductance [84], and accelerometry [85], show a strong relationship with human thermal sensation and comfort. Sim et al. [86] developed personal thermal sensation models based on wrist skin temperature measured by wearable sensors. In addition, studies using more than one category are also not uncommon. A "personalized" model can be developed by integrating the occupants' physiological and behavioral data. Other recent attempts [87] applied commercial wearable sensors together with environmental sensors (e.g., temperature, air speed) to predict the comfort of each individual occupant.

Even though all the above-reviewed studies claimed an enhanced prediction accuracy over conventional PMV and adaptive models, we identify three major drawbacks or limitations in those studies. First, subjects involved in the studies were restricted in a climate-controlled laboratory environment for a short period of time, usually a few hours [86]. The dynamics of thermal comfort among daily diverse activities (e.g., dining, commuting, working, shopping) and their interactions cannot be fully captured in steady-state short-term lab tests. Even in a relatively "static" office environment, occupants would be engaged in different tasks (e.g., attending meetings, working at computers, doing office chores). As such, studies at steady-state conditions could not capture human activity, circadian cycle and mobility. The feasibility and accuracy of personal thermal comfort models developed under real-life conditions are still unclear. From our literature review, the models developed directly from lab data [82] usually have higher prediction power (90% vs 70%) as compared to those from the real environment [75]. Second, most studies evaluated the performance of personal models, which predict categorical responses (e.g., cooler, warmer, no change), using accuracy that is the number of correctly predicted instances divided by the total number of instances in the dataset. Previous studies using such metric reported the prediction accuracy 79% $\pm$ 32% (Mean $\pm$ SD) for personal comfort models developed from physiological data with wearable sensors [75, 82]. However, this metric is problematic because it fails to exclude correct prediction purely due to randomness [88].

Third, previous studies with wearable sensors often employed commercialized low-cost sensors. The sensing accuracies of those sensors when they were worn were not known, even though manufacturers reported a high accuracy and strong reliance of the embedded sensors. In most situations, the manufacture specification was based on laboratory validation in a static environment, which could be quite different when sensors were used by end-users. For instance, the Empatica E4 (Empatica Inc., USA) wristbands or similar products might be only reliable with limited movements, for example, during sleeping and sitting at the table.

In the present study, to address the prior identified limitations, we developed personal

thermal comfort models by machine learning using lab grade wearable sensors that continuously monitor physiological signals (skin temperature, heart rate, accelerometry) for a long period in real settings. Since a personal comfort model applies individually relevant rather than group-averaged information for thermal comfort predictions, it can be better utilized to understand specific comfort needs and desires of individual occupants and satisfy their thermal comfort accordingly. With personal comfort models, a building system can provide optimal conditions for enhanced thermal satisfaction and energy efficiency. More practically, a personal model is able to evolve by adapting new data collected in future smart buildings. We aim to evaluate the prediction power of each personal comfort model using metrics that can compensate for randomness. The importance of physiological signals and environmental parameters for prediction were also assessed in this study.

2.2 Thermal Comfort Experiment

Unlike population-average models, a personal comfort model should be specifically developed for an individual occupant to account for great variations in personal factors. A personal model for an occupant might not be necessarily the same for another, even if its accuracy compared to a population-average model may be higher due to its flexibility. As such, personal models are determined using data-driven approaches such as continuous training of machine learning algorithms over streaming data [9]. In this study, we collected and formatted physiological responses from human subjects and then applied machine learning algorithms to train personal thermal comfort models for each subject. Thermal sensation and preference from surveys were utilized as ground truth for model training and evaluation. The following sub-sections describe our approach in detail.

Twenty subjects (half female and half male adults) living in Berkeley and San Francisco, CA, were initially recruited through posted announcements. We only analyzed the data from the fourteen subjects (6 female and 8 male adults) to develop personal thermal comfort models for each person. We asked each subject to take an online survey at least once every hour during the day. They were required to take the survey at least 12 times per day to capture the dynamics of thermal conditions, especially when their thermal sensations changed, such as after working out or moving to a different thermal environment. Developed with Qualtrics (Qualtrics, LLC), the survey included three "right-now" questions: (1) location (Indoor or Outdoor), (2) thermal sensation (continuous ASHRAE scale from -3 cold to 3 hot), and (3) thermal preference (Cooler, No change, and Warmer).

We also collected physiological signals in this study. We collected skin temperature at the wrist and ankle, heart rate, and wrist accelerometry. The physiological sensor set up is illustrated in Fig. 2.1.
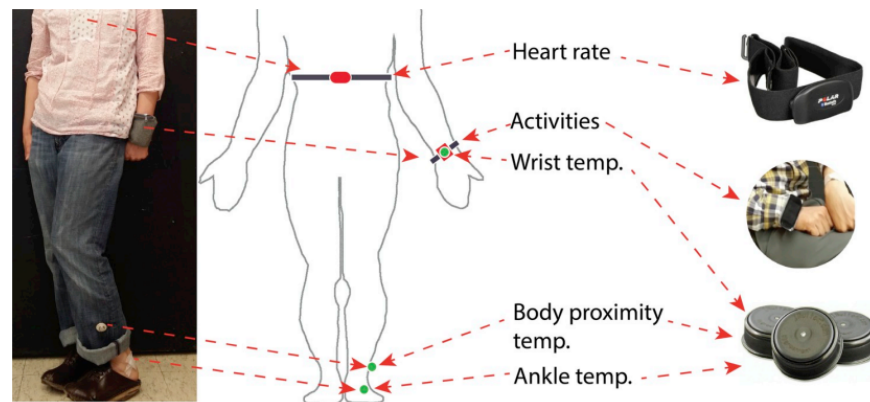
Figure 2.1: Physiological Sensor setup.

2.3   Data Analysis and Problem Identification

The features were extracted from raw data and consisted of skin temperatures (wrist and ankle), heart rate, body proximity temperature and weather conditions (wind, solar radiation, temperature, and humidity). We downloaded weather data from the station near each subject during the participation. People spend most of the time indoors, so one may think that these parameters are relevant only when people are outdoors. We argue that weather conditions may affect people's clothing choices, thermal expectation and, to a certain degree, the way how buildings are conditioned. These intermediate factors, which were not measured directly, may also cause variation of thermal perception and comfort.

For skin temperature and heart rate, we considered the average and gradient over the timeframes of 5 min and 60 min prior to a vote. The gradient was the slope of local linear regression (time vs variable) applied to the data within a timeframe window. A negative gradient of skin temperatures of the extremities possibly indicated a cool thermal sensation. Likewise, an increased (positive gradient) heart rate and body movement (inferred by measuring acceleration) might be associated with enhanced metabolism or energy expenditure. The standard deviation of acceleration suggested the intensity of a physical activity (e.g., walking). The selection of the time frame window was based on two assumptions. First, in most real-life situations, occupants' thermal conditions change little within 5 min. Second, physiological signals 60 min ago or earlier have little reflection on the present thermal conditions. The Pearson correlation coefficients between averaged heart rate over 5 min vs 15 min, 5 min vs 30 min, 5 min vs 60 min, 15 min vs 30 min, 15 min vs 60 min, and 30 min vs 60 min are 0.95, 0.92, 0.91, 0.96, 0.94, and 0.96, respectively. The high auto-correlations imply that finer or more timeframe windows might not be useful to improve prediction accuracy. Similar strong auto-correlations can be also found for skin temperatures of both wrist and ankle: 0.94 (5 min vs 15 min), 0.84 (5 min vs 30 min), 0.67 (5 min vs 60 min), 0.95 (15 min vs 30 min), and 0.92 (30 min vs 60 min). In addition, the time frame windows for the average
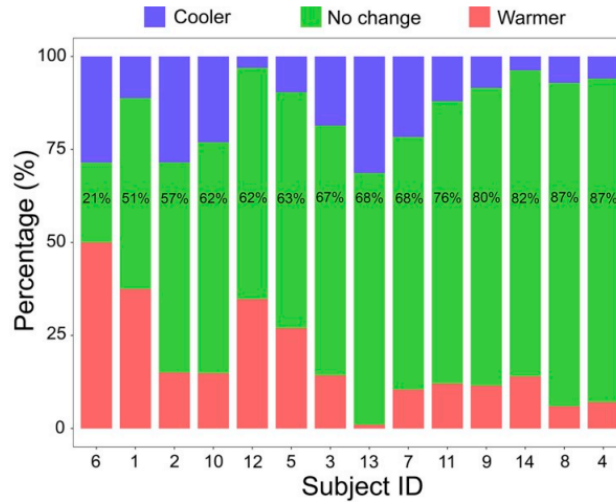
Figure 2.2: Distribution of Data Points for various Thermal Preference Classes

Table 2.1: Distribution of Number of Subjects in various Age Groups

Age Group	Number of Subjects
20-30	10
30-40	3
40-50	1
>50	0

and gradient of meteorological parameters are 1h and 8h.

The plots in Fig. 2.2 show the overall thermal sensation and preference (sorted by the percentage of "No change") for each subject. Consistent to a larger scale meta-data (the Comfort Database [54]), the median thermal sensations for all participants are within the thermal neutrality (-0.5 < Thermal Sensation (TS) < 0.5) except for the subject (ID = 5, TSmedian = 0.7) and the subject (ID = 6, TSmedian = -0.6). However, the range of thermal sensations varies extensively among them.

We observed that the distribution of data among the 3 thermal preference classes is highly imbalanced. This is expected since most of the systems in existing buildings are expected to achieve thermal neutrality. The distribution of individuals in various age groups, and various Body Mass Index (BMI) categories are provided in Table 2.1 and Table 2.2 respectively. We observe that the availability of data for extreme classes in each type, i.e. aged subjects, and overweight and obese subjects is rare. This can be partly attributed to the difficulty in logistics of conducting experiments with aged or frail subjects. This class imbalance in the nature of the data causes challenges in ML-based modeling for thermal comfort prediction.

Table 2.2: Distribution of Number of Subjects in various BMI Groups

Age Group	Number of Subjects
Underweight (<19)	9
Healthy (19-25)	3
Overweight (25-30)	2
Obese (>30)	0

2.4 Machine Learning based Thermal Preference Prediction

Model Selection

We applied various machine learning algorithms to develop personal thermal comfort models over the collected dataset. The predicted response of the models was thermal preference ("Cooler", "No change" or "Warmer") because it is the most relevant parameter addressing thermal discomfort by specifying which action a heating, ventilation, and air conditioning (HVAC) system should take. The dataset consisted of numerical variables mainly measured by wearable sensors and subjective votes that were numerical (e.g., thermal sensation) or categorical (e.g., thermal preference). We applied four groups of machine learning algorithms (1) linear methods, (2) non-linear methods, (3) trees and rules, and (4) ensembles of trees with each including several commonly used classification algorithms. This algorithm selection ensures that prediction biases can be well balanced, preventing over- or under-prediction resulting from specific algorithms. Each algorithm can be applied to train a personal thermal comfort model based on the data-driven method.

Evaluation Metrics

The performance of all the developed personal thermal comfort models from the machine learning algorithms was evaluated by three commonly used metrics: Cohen's kappa [89] (measures the agreement between two raters who each classify the items into some mutually exclusive categories), accuracy [75], and Area Under the Receiver Operating Characteristic (ROC) Curve (AUC).

2.5 Results

Table 2.3 summarizes prediction performance with the metrics of Cohen's kappa/accuracy/AUC using 14 machine-learning algorithms for all participants. For all these subjects, the median prediction Cohen's kappa of personal comfort models is 20% with coincidence accuracy of 68% and AUC of 0.69. When only the best performing algorithm for each subject is considered, the

Table 2.3: Prediction power (Cohen's kappa/accuracy/AUC) for each participant with 14 common machine learning algorithms.

SubID/Data Size	1/152	2/253	3/323	4/261	5/271	6/242	7/393
Lda	21%/56%/0.66	24%/61%/0.7	45%/74%/0.79	2%/83%/0.53	37%/69%/0.77	11%/48%/0.58	20%/68%/0.69
regLogistic	17%/56%/0.68	22%/62%/0.73	40%/75%/0.82	0%/87%/0.57	30%/69%/0.79	15%/53%/0.64	15%/70%/0.7
nnet	21%/56%/0.68	26%/62%/0.73	50%/77%/0.86	7%/78%/0.62	38%/70%/0.79	15%/51%/0.62	18%/68%/0.7
svmRadial	15%/54%/0.58	19%/60%/0.72	44%/76%/0.84	0%/87%/0.64	32%/70%/0.76	17%/55%/0.61	13%/69%/0.67
knn	11%/52%/0.61	24%/59%/0.71	43%/76%/0.83	1%/86%/0.65	29%/69%/0.76	15%/51%/0.59	17%/68%/0.62
nb	13%/49%/0.62	22%/55%/0.7	47%/72%/0.81	5%/75%/0.6	32%/63%/0.74	14%/45%/0.6	21%/65%/0.65
rpart	7%/49%/0.55	45%/71%/0.65	31%/71%/0.74	3%/85%/0.55	30%/68%/0.66	10%/50%/0.56	17%/66%/0.58
J48	7%/46%/0.55	52%/72%/0.61	40%/71%/0.7	9%/82%/0.53	31%/68%/0.66	10%/48%/0.55	13%/69%/0.54
PART	9%/47%/0.56	43%/68%/0.63	39%/71%/0.72	8%/82%/0.54	29%/64%/0.65	12%/46%/0.56	13%/61%/0.59
C5.0	10%/50%/0.61	65%/80%/0.73	47%/75%/0.85	6%/86%/0.63	32%/66%/0.78	21%/53%/0.58	19%/66%/0.65
treebag	11%/51%/0.61	49%/73%/0.73	46%/75%/0.84	6%/85%/0.65	34%/68%/0.76	16%/51%/0.6	17%/68%/0.66
gbm	19%/55%/0.67	54%/75%/0.78	50%/77%/0.85	7%/85%/0.68	4%/71%/0.8	16%/52%/0.62	20%/69%/0.68
extraTrees	19%/57%/0.67	51%/74%/0.78	50%/78%/0.88	7%/86%/0.73	37%/70%/0.8	17%/53%/0.63	21%/70%/0.7
rf	17%/55%/0.64	51%/74%/0.75	48%/76%/0.86	7%/87%/0.68	34%/68%/0.8	18%/54%/0.62	18%/69%/0.68

SubID/Data Size	8/353	9/261	10/256	11/399	12/164	13/198	14/322
Lda	40%/87%/0.76	18%/77%/0.68	17%/62%/0.65	34%/79%/0.74	22%/64%/0.71	33%/71%/0.66	8%/80%/0.7
regLogistic	6%/87%/0.76	2%/80%/0.7	7%/62%/0.69	22%/79%/0.76	8%/62%/0.74	41%/75%/0.83	2%/82%/0.71
nnet	34%/85%/0.78	20%/74%/0.72	21%/64%/0.68	31%/79%/0.77	16%/62%/0.74	37%/73%/0.76	11%/78%/0.72
svmRadial	25%/88%/0.82	4%/79%/0.72	5%/62%/0.64	32%/80%/0.76	0%/60%/0.69	35%/75%/0.78	1%/82%/0.73
knn	9%/87%/0.75	15%/76%/0.73	17%/60%/0.66	18%/77%/0.7	6%/59%/0.63	40%/75%/0.7	0%/82%/0.68
nb	35%/84%/0.79	20%/70%/0.69	16%/48%/0.66	32%/71%/0.76	24%/62%/0.72	41%/73%/0.78	16%/74%/0.72
rpart	19%/84%/0.59	16%/75%/0.64	12%/61%/0.58	24%/76%/0.68	11%/60%/0.55	34%/72%/0.66	6%/75%/0.55
J48	27%/84%/0.58	22%/74%/0.61	11%/61%/0.56	20%/72%/0.63	19%/62%/0.64	37%/74%/0.65	6%/77%/0.57
PART	27%/85%/0.58	21%/75%/0.62	13%/55%/0.57	21%/72%/0.65	15%/62%/0.64	35%/72%/0.62	5%/77%/0.58
C5.0	27%/88%/0.8	23%/78%/0.75	16%/59%/0.65	33%/77%/0.76	19%/63%/0.67	38%/74%/0.69	6%/77%/0.66
treebag	30%/87%/0.78	21%/79%/0.77	18%/63%/0.65	30%/78%/0.76	18%/63%/0.71	42%/75%/0.78	3%/79%/0.66
gbm	31%/88%/0.79	24%/78%/0.79	15%/61%/0.67	37%/79%/0.79	19%/63%/0.74	47%/77%/0.81	6%/79%/0.71
extraTrees	33%/88%/0.84	21%/79%/0.81	20%/64%/0.7	32%/80%/0.8	18%/63%/0.76	46%/78%/0.81	4%/80%/0.75
rf	29%/87%/0.81	23%/79%/0.8	18%/63%/0.67	33%/79%/0.78	18%/64%/0.76	45%/76%/0.8	1%/80%/0.7

median (based on kappa) prediction power is 24%/78%/0.79 (Cohen's kappa/accuracy/AUC). Kim et al. [9] reported the median prediction AUC of personal models, 0.73, by analyzing the heating and cooling behavior of 34 out of 38 occupants in an office building. The results show that prediction power fluctuates among subjects and algorithms. The personal thermal comfort model of Subject 2 shows the highest median prediction power (44%/69%/0.73). By contrast, the model of Subject 4 displays the weakest performance (6%/85%/0.62), almost random "guessing" in terms of the low Cohen's kappa. Worthy of notice here is that the accuracy would be misleading (Subject 4 has a higher accuracy than Subject 2) because of the imbalanced dataset. Subject 4's data were probably problematic, because we found that the subject always responded with "No change" for three consecutive days. In addition, preferring "Warmer" while feeling warm (TS = 1.4), and "Cooler" while cold (TS = -1.5), existed in the survey answers. As thermal comfort is subjective, we cannot conclude that those data are faulty. However, we can hypothesize that the subject did not answer carefully, and this could be a reason for the low prediction power. Another possibility is that some people might be less predictable than others.

2.6 Towards Neural Network based Modeling

Neural networks can have significant potential both in better modeling of personal thermal comfort, and to introduce use of several advanced algorithms. In a study conducted by the University of Pennsylvania in 2021, researchers used a Bayesian Neural Network (BNN) to predict a room occupant's thermal preferences. They chose this architecture because the Bayesian method utilizes both prior knowledge of occupants' preferences as well as real-time measurements for prediction [90]. A study conducted at George Washington University utilized thermal sensors to extract temperature information from subjects and machine learning to predict the thermal comfort of individuals to adjust their Heating, Ventilation, and Air Conditioning (HVAC) systems [91].

In this part, we aimed to build on these prior studies, but instead of using kernel-based algorithms for thermal comfort prediction, we propose time-series modeling using deep-learning for the same purpose. Given a series of observations $x^{(1)}, \cdots, x^{(t)}$ we generate $y^{(t+1)}$, our hypothesis for the next label. Furthermore, we do it at an individual level, i.e. personal thermal comfort models.

Deep learning methods are powerful tools for time series analysis as they can extract high level patterns using non-linearities. Such methods include artificial neural networks, convolutional neural networks and recurrent neural networks. RNNs are often used in the context of time series prediction for the ability to leverage temporal relationships. However, one problem that arises from the unfolding of an RNN is that the gradient of some of the weights starts to become too small if the network is unfolded for too many time steps. This is called the vanishing gradient problem. A type of network architecture that solves this problem is the LSTM, which utilizes a unique gradient structure that ensures there is at least one path where the gradient does not vanish.

In many contexts, however, overfitting is a major issue in creating well-generalizable models for time series applications. In this regard, we use a novel regularized LSTM (R-LSTM) as a way to develop personal thermal comfort models for application in smart building contexts. To the best of our knowledge, there is limited work utilizing regularized time-series based models for thermal comfort prediction in literature.

Data Pre-processing

The quantity of the initial, unfiltered survey data and physical data for each participant can be seen in Fig. 2.3. We separated both the survey, physiological, and environmental data by each participant. For each given participant, we filtered the physiological and environmental data, removing all rows whose time stamps were not within 30 minutes of at least one timestamp from the subject's corresponding survey data. We then merged that subject's physical and survey data and imputed any missing values using forward filling and then backward filling. Because every survey timestamp corresponded to exactly one timestamp of the physical data, we ensured that all the imputed data were within a 30-minute range of the
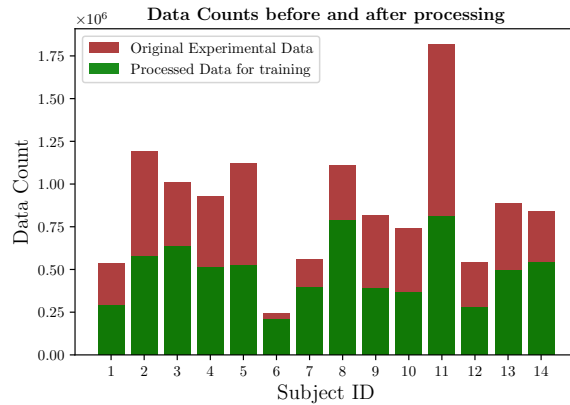
Figure 2.3: Data Counts before and after processing

data at that imputed time step. The number of samples of filtered data for each participant can also be seen in Fig. 2.3.

A significant challenge was working with the large proportion of missing data and irregularly sampled observations. For instance, features like heart rate had a higher frequency than external humidity. The data was re-sampled on a minute-by-minute basis, using the mean value for feature x_i during each period. To account for gaps between sections of recorded time (for instance, between the end of day i and the start of day $i + k, k > 0$), a delta time feature was incorporated. However, no feature aggregation and further processing was done on the existing time series columns, in contrast to previous ML approaches.

For the time-series task, 3D data were generated from the dataset using a sliding window, representing the number of time steps used to 'look back' before predicting the next outcome. In this work, a period of 120 time-steps were used. This number was chosen from small pilot tests (with subsets of the processed dataset) and with the pragmatic consideration of roughly two hours providing adequate information to make a prediction.

Modeling

The model architecture consisted of a core LSTM [92] layer connected to a dropout layer, and a dense layer with the softmax activation function applied to produce three class probabilities, corresponding to a predicted thermal preference of "cooler", "no change", and "warmer". We use Adam optimizer, with varying decay rates.

The regularized loss has a l_1 regularization. The l_1 norm was chosen for its ability to improve sparsity, resulting in feature subset solution and sparser optimization solutions.

Metric/Experiments

To evaluate the model on different hyper-parameters and architectures, the categorical accuracy and weighted average F1 were chosen as metrics. It was also found that many subjects had severe class imbalance, so a normalized weighting was applied to data corresponding to each of the three class labels.

To evaluate experiments, 5-fold time-series cross-validation was employed, consisting of increasing super-sets of training data. This is required as there is a temporal dependency between observations, and we must preserve that relation during testing.

Results

The results from best performing R-LSTM models for each subject are included in Table 2.4. On average for the different subject datasets, the model achieved an accuracy of 78%, weighted F1 of 0.74, and AUC.

Overall, the R-LSTM showed promising results on the time series data, and had the crucial advantage of minimizing extra feature engineering (through the aggregation of different physical signals over a fixed period of time).While there was variation between datasets, it was found that the large batch size (up to 512) tended to produce better results, especially when paired with relatively higher learning rates. This reflects the ability for the model to escape from local minima found from noise that plays a more significant role with smaller batch sizes (approaching that of stochastic gradient descent). Furthermore, the model often found lead-in periods of comfort preference where long sections of an outcome (such as prefer "warmer"followed by "no-change") and performed well when such patterns were present in the validation data.

Overfitting proved to be a major issue for this task, so both dropout regularization and the l_1-regularization were important to ensure the model still had sufficient capacity without memorizing the training data. This was particularly salient when the data was extremely imbalanced, despite the weighted resampling method that was used. Another difficulty was stabilizing the training process without getting trapped in local minima and inconsistency between datasets. In general, the cross-folds with more data showed better results.

2.7 Conclusions

The thermal comfort of individuals can have a strong impact on health and well-being. Especially in tropical, humid countries, it is important to try to understand the deleterious effects that environmental exposures can have on the built environment. In order to eventually better understand the physiological basis of thermal preferences of individuals, we built a time-series based deep learning model for thermal comfort prediction.

Predicting thermal comfort/preference using physiological data could be potentially incorporated into HVAC system control for occupants' satisfaction and energy saving. The low-cost wearable sensors and cloud computing allow real-time thermal comfort/preference

Table 2.4: Thermal Preference Prediction results using state-of-the-art random forest model vs time-series based LSTM model.

Subject ID/ Survey Data Size	State-of-the-art Random Forest Model (Accuracy/F1/AUC)	Time-Series based LSTM Model (Accuracy/F1/AUC)
1/152	55%/-/0.64	82%/0.78/0.76
2/253	74%/-/0.75	79%/0.79/0.74
3/323	76%/-/0.86	80%/0.75/0.75
4/261	87%/-/0.68	85%/0.83/0.84
5/271	68%/-/0.8	75%/0.76/0.74
6/242	54%/-/0.62	66%/0.65/0.65
7/393	69%/-/0.68	76%/0.75/0.74
8/353	87%/-/0.81	82%/0.77/0.76
9/261	79%/-/0.8	84%/0.76/0.80
10/256	63%/-/0.67	76%/0.69/0.70
11/399	79%/-/0.78	68%/0.66/0.69
12/164	64%/-/0.76	76%/0.76/0.74
13/198	76%/-/0.8	73%/0.74/0.72
14/322	80%/-/0.7	86%/0.81/0.82

prediction using physiological and environmental data. We developed personal thermal comfort models for 14 participants using lab-grade wearable sensors. Based on physiological and meteorological data monitored for 2–4 weeks, we trained 14 personal comfort models using different machine learning algorithms for each participant.

We also applied deep learning architecture to the prediction of thermal comfort preferences for 14 individuals. Our deep learning model is a novel approach to thermal comfort prediction by combining l_1 regularization with an R-LSTM. We found that our deep-learning model was successful in comparison to state-of-the-art kernel-based machine learning models for thermal comfort prediction.

Some challenges existing in the dataset was also identified, including class imbalance problem in thermal preference data distribution, and scarcity of data for extreme cases as per age, and body mass index types. In the future chapters, we will design machine learning based solutions to tackle these challenges.

Chapter 3

Energy Game-Theoretic Frameworks and Segmentation Analysis

3.1 Introduction

Energy game-theoretic frameworks can analyze occupant behavior and possibly modify it, by engaging the users in the process of energy management, integrating cyber-physical technology and by leveraging humans-in-the-loop strategy [48, 93–96]. Such game-theoretic frameworks can be thought of as a sensor-actuator system. Through their participation in the game (the sensor), the behavior of users is observed, which then is treated as the input to an incentive design process (the actuator). The incentives offered can motivate users to improve upon their energy usage behaviors in order to achieve better energy efficiency, signifying the importance of an intelligent incentive design in the success of such frameworks. Although all such frameworks aim to achieve a long term or permanent improvement in the energy usage behaviors among the users, the aim is seldom achieved after the completion of the energy game, mostly attributed to the lack of an intelligent and adaptive incentive design process. The incentive design process in prior works is dependent on utility functions of every player in the game, which is hard to compute as energy game-theoretic frameworks involve participation of a large number of energy users. Instead, the utility/energy usage behavior of a large number of players can be simplified by grouping the players into a relatively small number of clusters of similar behavior. Incentives can then be designed to tailor each cluster assuming players in a cluster have similar behavior. Energy utility companies frequently use such segmentation techniques for optimal planning of demand response, load shedding, and microgrid applications [97]. In this work, we consider the design of a smarter segmentation analysis as a solution for intelligent incentive design for energy game-theoretic frameworks.

This can be achieved by learning the factors leading to human decision-making, and using the knowledge to devise a novel agent segmentation method. The segmentation analysis in an energy game-theoretic framework with high dimensional data requires powerful yet computationally efficient statistical methods. A possible candidate, Graphical Lasso algorithm

[98], has been widely applied on different scientific studies due to its sparsity property (ℓ_1 penalty term) and efficiency [99, 100]. The potential of Graphical Lasso can be innovatively combined with player segmentation. Towards this, we enable new avenues by combining both concepts and applying it on a social game data set [93] to classify the energy efficiency behaviors among building occupants. We explore the causal relationship between different features of the agents using a versatile tool, Granger's Causality [101], which leads to a deep understanding of decision-making patterns and helps in integrating explainable game theory models with adaptive control or online incentive design. We propose an explainable, rather than just a black box model. To summarize, our contributions are threefold:

- Novel segmentation analysis using an explainable statistical model at the core towards learning agent's (building occupants) their decision-making in competitive environments [102, 103].

- Characterization of causal relationship among several contributing features explaining decision-making patterns in agent's actions.

- Improving building energy efficiency by using the proposed segmentation analysis method.

3.2   Related Work

Energy game theoretic frameworks have enabled an effective platform for incorporating energy efficient behavior among the building occupants in a smart building. They involve the important aspect of human participation in building control, otherwise lacking in many conventional modeling approaches including passive Hidden Markov Models (HMMs) [104]. A number of such frameworks have been introduced over the years [93, 105, 106], which have shown significant post game energy reduction.

For incentive design, many game-theoretic frameworks [107–109] rely on knowledge of utility functions of the players in the game, which are hard to compute in the scenario of energy game theoretic frameworks due to the complexity and scale. Authors in [110] propose a Nash-equillibrium based approach for utility estimation. In [111], authors formulate the utility estimation problem as a convex optimization problem by using first-order necessary conditions for nash equilibria, and then create an affine map along with energy consumption to derive the utilities. All these methods are hard to scale when the number of players is high. Instead, we can segment the utilities of players into clusters by learning features characterizing human decision-making in competitive environments, and performing an incentive design for the clusters so obtained. We derive inspiration for agent segmentation owing to the fact that customer segmentation has been successfully utilized in energy systems [97]. The energy usage behavior exhibited by each player in a cluster is expected to be similar, and, statistically, it has been shown that a relatively small number of clusters is adequate in describing a wide scope of customer behaviors.
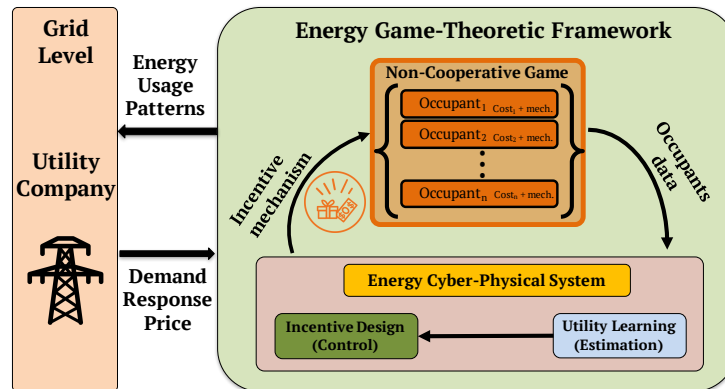
Figure 3.1: Gamification abstraction of the Energy Social Game acting as our data source.

Towards this we use high dimensional real-world data. We use the graphical lasso algorithm as a powerful tool to understand the latent conditional dependence between variables [98]. This in turn provides insights into how different features interplay among each other. Historically, Graphical Lasso has been used in various fields of science, ranging from study of how individual elements of the cell interact with each other [99] and to the broad area of computer vision for scene labelling [100]. A modified version of the original algorithm, named time-varying graphical lasso, has been used on financial and automotive data [112]. However, the novelties of graphical lasso has not been well utilized in the area of energy cyber-physical systems. We use Granger's causality [101] to explain the causal relationship between the features in energy usage behavior of agents in social game. It has been widely used in the energy domain in applications such as deducing the causal relationship between economic growth and energy consumption [113].

Knitting novel segmentation algorithms and their application to energy game-theoretic frameworks together, we employ graphical lasso algorithm for customer segmentation on social game dataset and present an explainable model, helpful both in understanding inherent factors leading to energy efficiency in buildings and in intelligent incentive design [114,115].

3.3 Methods

Energy Game-Theoretic Dataset

The dataset used for our work is from an energy social game experiment to encourage energy efficient resource consumption in a smart residential housing, as introduced in [93]. Authors in [93] designed a social game among occupants of residential student housing apartments at an university campus (Fig. 3.1). They make use of Internet of Things (IoT) sensors to allow the occupants to monitor their room's lighting (desk and ceiling light) and ceiling fan usage via a personal web-portal account as they participate in the energy social game

for maximizing their incentives. The above game-theoretic framework is modelled under the umbrella of a multiplayer non-cooperative game. The dataset consists of per-minute time-stamped reading of each resource's (desk light, ceiling light and ceiling fan) status, accumulated resource usage (in minutes) per day, resource baseline, gathered points (both from game and surveys), occupant rank in the game over time and number of occupant's visits to the web portal. It also contains features related to time of day (morning vs. evening), time of week (weekday vs. weekend) and college schedule feature indicators for dates related to breaks, holidays, midterm and final exam periods. Additionally, the dataset incorporates the external weather metrics during the experimental run.

Trade-off between Supervised/Unsupervised Segmentation

For the purpose of segmentation analysis, both supervised and unsupervised segmentation methods can be implemented on the social game dataset. Supervised methods require a label to classify data with similar labels together. For the dataset in hand, the label we have is the rank of the player in the game, which in turn indicates their energy efficiency characteristics as compared to other players in the game (i.e. a player with less rank is more energy efficient). We use rank as the label to classify players into different groups. But, such a classification method groups different players together as per their overall rank, and does not take into account the distribution of their energy efficiency characteristics across different scenarios such as time. For instance, Figure 3.2 shows the distribution of cumulative resource usage (in minutes/day) for a player having low rank (high energy efficiency) and a player having high rank (low energy efficiency), with some curve smoothing across a duration of the game period. It can be observed that the high energy efficient player performs sub-optimally (uses more energy resources) between the times A and B than the low energy efficient player. In an ideal scenario, for every player, the data samples corresponding to low energy efficient behavior should be clustered separately than high energy efficient behaviors so as to have an accurate understanding of the interplay of features governing human decisions for energy efficiency. In this case, unsupervised clustering proves helpful and clusters together similar behaviors. But in this case, the output of unsupervised clustering is just a number of clusters with no labelling about the energy efficiency characteristics exhibited by that cluster. So, to summarize, supervised classification provides insight into an overall picture of how different classes of energy-efficient players behave, but fails to capture the distribution of behaviors. On the other hand, unsupervised clustering captures the latter accurately, but does not provide any information on labels of the clusters generated. This poses a trade-off between supervised classification and unsupervised clustering methods for application in energy games.

Proposed Segmentation Method

The trade-off mentioned in previous section signals to use the novelty of both unsupervised and supervised segmentation together to build an optimal model. Knitting together both the
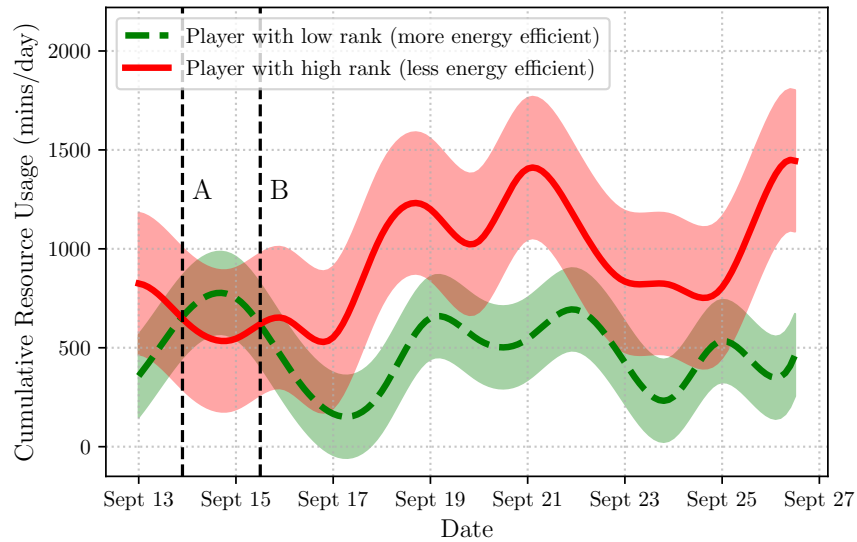
Figure 3.2: Variation of cumulative energy resource usage (mins/day) for a player with low rank (high energy efficient) and another with high rank (low energy efficient)

methods via a powerful tool, the graphical lasso algorithm, we present a novel methodology to perform segmentation in energy game-theoretic frameworks. We employ the k-means algorithm for unsupervised clustering. First, the optimal number of clusters in the dataset is derived using elbow method and silhouette scores. An elbow plot is a plot between the distortion score (a measure of closeness of data points to their assigned cluster center) vs the number of clusters. The optimal number of clusters is determined to be corresponding to drastic change in the rate of reduction in distortion score. The elbow plot for energy social game dataset, obtained in an unsupervised manner is given in Figure 3.3. The optimal number of clusters is determined to be 3. We also use another metric, the silhouette score to confirm the optimal number of clusters. The silhouette score $\in [-1, 1]$, is a measure of how similar an object is to its own cluster compared to other clusters. A high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. The silhouette score corresponging to each number of clusters is given in Table 3.1. Note that the score is the highest (shaded in blue) for the number of clusters as 3. Following this, we use

No. of Clusters	2	3	4	5
Silhouette Scores	0.684	0.749	0.611	0.540

Table 3.1: Silhouette Scores for different number of clusters

Minibatch k-means algorithm with k= 3 (optimal number of clusters in the data) to obtain the clusters. Let the clusters obtained be represented by C_{unsup}^1, C_{unsup}^2 and C_{unsup}^3. Since

Distortion score vs number of clusters(k) for K-means clustering
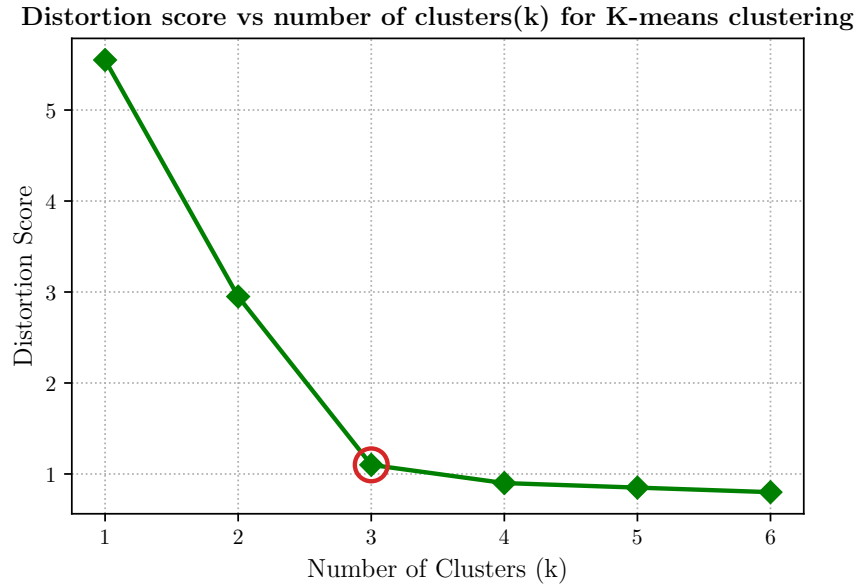
Figure 3.3: Elbow Plot to choose optimal number of clusters
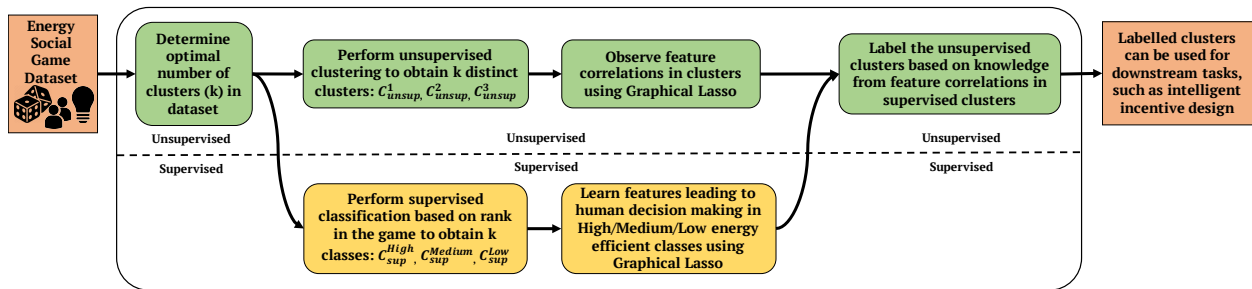
Figure 3.4: Overview of the proposed segmentation method

the dataset correspond to energy usage behavior of the players, the three clusters so obtained correspond to high, medium and low energy efficient behaviors. We then use supervised classification and graphical lasso to label the unsupervised clusters. We divide the players into three classes in a supervised way taking the ranks of the users as the label.

Let the players be denoted by $P_1, P_2, \cdots, P_m$ and the data points corresponding to the ith player across time be $d_1^i, d_2^i, \cdots, d_{n_i}^i$. The whole range of ranks were divided into three equal segments, with the high, medium and low energy efficient rank groups being R_{High}, R_{Medium} and R_{Low} respectively. Let the classes be represented by C_{sup}^{High}, C_{sup}^{Medium} and C_{sup}^{Low}, where the superscripts signify the energy efficiency behavior of each class. We assign the

players to the classes as per the following formula, $P_i \in C_{sup}^X$, where,

$$X = \underset{x \in [low, medium, high]}{argmax} \left\{ \sum_{j=1}^{n_i} \mathbb{1}[rank(d_j^i) \in R_x] \right\} \tag{3.1}$$

where $\mathbb{1}[\,\cdot\,]$ is the indicator function. This allocates each player into one of the three supervised classes. The behavior of a player in a particular class, e.g. C_{sup}^{High} represents the characteristic behavior of players showcasing high energy efficiency. Then the feature correlations in all the supervised classes and unsupervised clusters were studied using graphical lasso algorithm. Knowledge of feature correlation similarity among members of the supervised classes and unsupervised clusters is used to label the unsupervised clusters (C_{unsup}^1, C_{unsup}^2 and C_{unsup}^3) as high/medium/low energy efficient. Finally, the labelled unsupervised clusters can be further explored for downstream tasks, such as incentive design. The whole process is illustrated in Figure 3.4.

3.4 Graphical Lasso for Energy Social Game

In this section, we formulate a framework towards segmentation analysis that allows us to understand the users decision making model. Specifically, we adopt graphical lasso algorithm [98, 116] to study the way in which features in an energy game-theoretic framework interplay among each other.

Let the features representing the social game data be denoted by the collection $Y = (Y_1, Y_2, \cdots, Y_S)$. From a graphical perspective, Y can be associated with the vertex set $V = \{1, 2, \cdots, S\}$ of some underlying graph. The structure of the graph is utilized to derive inferences about the relationship between the features. We use the graphical lasso algorithm [98] to realize the underlying graph structure, under the assumption that the distribution of the random variables is Gaussian.

Consider the random variable Y_s at $s \in V$. We use the Neighbourhood-Based Likelihood for graphical representation of multivariate Gaussian random variables. Let the edge set of the graph be given by $E \subset V \times V$. The neighbourhood set of Y_s is defined by

$$\mathcal{N}(s) = \{k \in V | (k, s) \in E\} \tag{3.2}$$

and the collection of all other random variables are represented by:

$$Y_{V \setminus \{s\}} = \{Y_k, k \in (V - \{s\})\} \tag{3.3}$$

For undirected graphical models, node for Y_s is conditionally independent of nodes not directly connected to it given $Y_{\mathcal{N}(s)}$, i.e.

$$(Y_s | Y_{V \setminus \{s\}}) \sim (Y_s | Y_{\mathcal{N}(s)}) \tag{3.4}$$

The problem of constructing the inherent graph out of observations is equivalent to finding the edge set for every node. This problem becomes predicting the value of Y_s given $Y_{\mathcal{N}(s)}$, or equivalently, predicting the value of Y_s given $Y_{V\setminus\{s\}}$ by the conditional independence property. The conditional distribution of Y_s given $Y_{V\setminus\{s\}}$ is also Gaussian, so the best predictor for Y_s can be written as:

$$Y_s = Y_{V\setminus s}^T . \beta^s + W_{V\setminus s} \tag{3.5}$$

where $W_{V\setminus s}$ is zero-mean gaussian prediction error. The β^s terms dictate the edge set for node s in the graph. We use l_1-regularized likelihood methods for getting a sparse β^s. Let the total number of data samples available be N. The optimization problem is formulated as: corresponding to each vertex $s = 1, 2, \cdots, S$, solve the following lasso problem:

$$\hat{\beta^s} \in \underset{\beta^s \in \mathbb{R}^{S-1}}{argmin} \left\{ \frac{1}{2N} \sum_{j=1}^{N} (y_{js} - y_{j,V\setminus s}^T \beta^s)^2 + \lambda\|\beta^s\|_1 \right\} \tag{3.6}$$

——— Algorithm 1: Graphical Lasso Algorithm for Gaussian Graphical Models ———

1. For vertices $s = 1, 2, \cdots, S$:

    a) Calculate initial loss $\|Y_s - Y_{V\setminus s}^T \beta^s\|_2^2$

    b) Untill Convergence:

        i. Calculate partial residual $r^{(s)} = Y_s$ - $Y_{V\setminus s}^T \beta^s$
        ii. For all $j \in V\setminus s$, Get $\beta_j^{s,new} = \mathbf{S}_\lambda\left(\frac{1}{N}\langle r^{(s)}, Y_j\rangle\right)$
        iii. Compute new loss $= \|Y_s - Y_{V\setminus s}^T \beta^{s,new}\|_2^2$
        iv. Update $\beta^s = \beta^{s,new}$

    c) Get the neighbourhood set $\mathcal{N}$(s)= supp(β^s) for s

2. Combine the neighbourhood estimates to form a graph estimate $G = (V, E)$ of the random variables.

$\mathbf{S}_\lambda(\theta)$ is soft thresholding operator as $sign(\theta)(|\theta| - \lambda)_+$.

For optimal design of penalty factor λ in Graphical Lasso run for a vertex s, we take 10 values in logarithmic scale between λ_{max} and λ_{min} as given below and conduct a line search to find the penalty factor which brings the minimum loss.

$$\lambda_{max} = \frac{1}{N} \underset{j\in V\setminus s}{max}|\langle Y_j, Y_s\rangle|, \quad \lambda_{min} = \frac{\lambda_{max}}{100} \tag{3.7}$$

Implementing a coordinate descent approach [98], the time complexity of the proposed algorithm is $O(SN)$ for a complete run through all S features. We also do 5-fold cross validation to ensure accurate value of the coefficients β^s. Use of partial residuals for each node significantly reduces the time complexity of the algorithm.

3.5 Results

As has been introduced in Section 3.3, we learn the feature correlations using graphical lasso algorithm in supervised classes C_{sup}^{High}, C_{sup}^{Medium} and C_{sup}^{Low} to obtain the knowledge about factors governing human decision making towards various (high/medium/low) energy efficient behaviors.

Feature correlation learning in supervised segregation

We consider a representative player (selected as the player holding the median rank in the class) for each of the three classes obtained out of supervised segregation method described in Section 3.3 to run graphical lasso and study the correlation between the features for that class. We group the features into different categories so as to study their influence on energy efficiency behaviors. Specifically, we consider *Temporal* features like time of the day, academic schedules and weekday/weekends, *External* features as outdoor temperature, humidity, rain rate etc. and *Game Engagement* features like frequency of visits to game web portal.

The feature correlations for a low energy efficient player is given in Fig 3.5. The player tries to use each resource independently which can be observed in Figure 3.5(a) with no correlation between the corresponding resource usage identifiers. There is a positive correlation between morning and desk light usage indicating heedless behavior towards energy savings. The absolute energy savings increase during the breaks and finals, given by positive correlation with total points, but it is not significant as compared to other players during the same period, thus increasing the rank. External parameters play a significant role in energy usage behavior of this class. The operation of the ceiling fan is driven by external humidity as given in Figure 3.5(b). Figure 3.5(c) indicates that their frequency of visits to the game web portal is motivated by sub-optimal performance in the game.
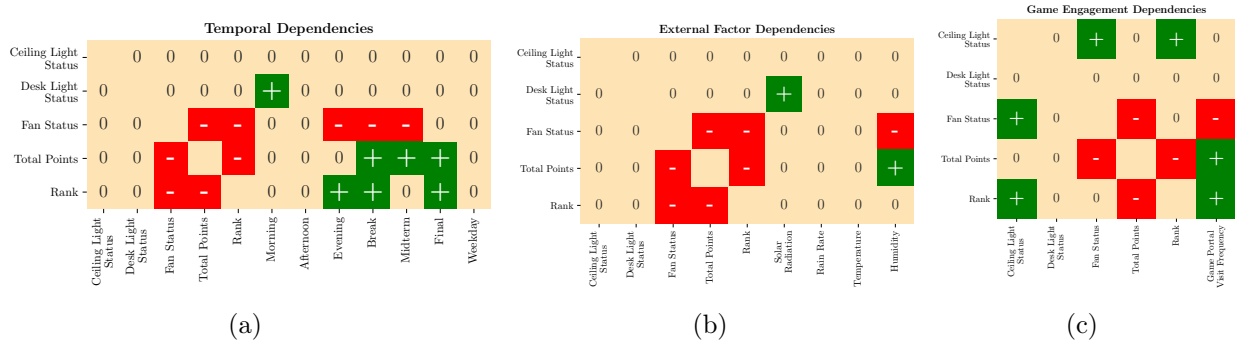
Figure 3.5: Feature correlations for a Low Energy Efficient Player ($\in C_{sup}^{Low}$)
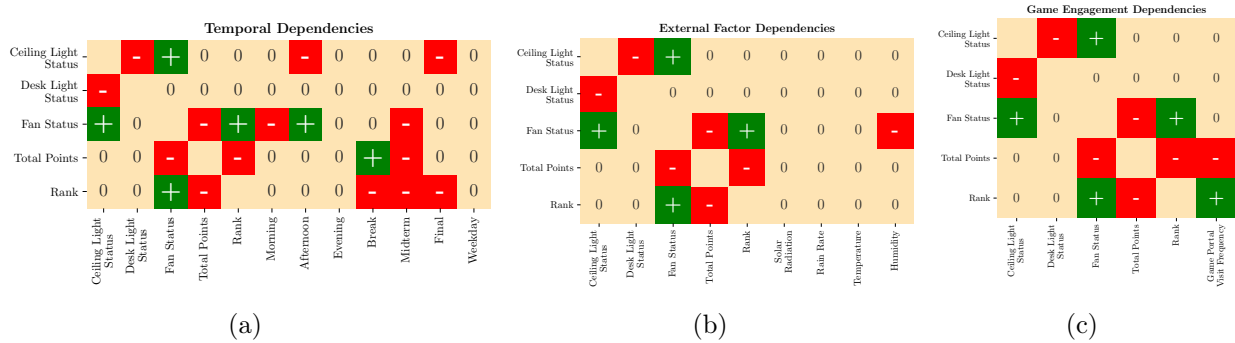
Figure 3.6: Feature correlations for a Medium Energy Efficient Player ($\in C_{sup}^{Medium}$)
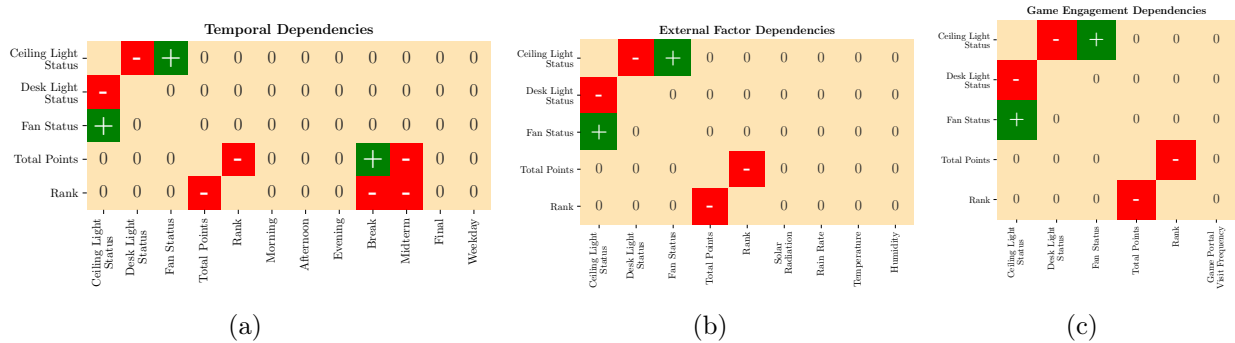
Figure 3.7: Feature correlations for a High Energy Efficient Player ($\in C_{sup}^{High}$)
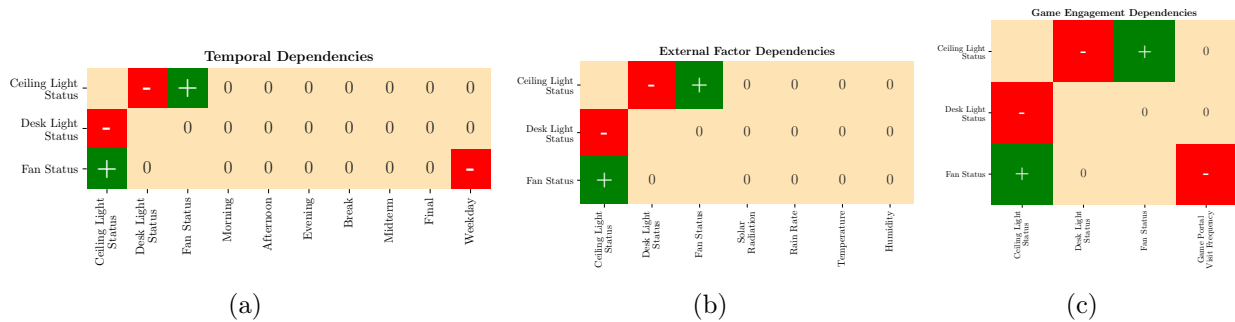
Figure 3.8: Feature correlations for energy usage behaviors in C_{unsup}^{3}. The labels "Total Points" and "Rank" are removed for unsupervised clustering.

Feature correlations for a medium energy efficient player is given in Fig 3.6. The player showcases predictable behaviors with correlations between desk light, ceiling light and ceiling fan usage (Figure 3.6(a)). The player co-optimizes the usage by alternating the use of ceiling and desk light. Different occasions like *break, midterm* and *final* are marked by energy saving patterns. Unlike a low energy efficient player, the player in this class tries to save energy in a conscious manner shown by reduced fan usage during the morning and reduced light usage during the afternoon. The fan usage is influenced by the external humidity, shown by Fig 3.6(b). The game engagement patterns for a player in this class (Fig 3.6(c)) is similar to that of the low energy efficient class.

Fig 3.7 shows the feature correlations for a high energy efficient player. This player also exhibits predictable behavior. Opportunistically, this player saves energy during breaks and midterms as shown by negative correlation between the corresponding flags and rank in Figure 3.7(a). Notice that there exists a negative correlation between midterm flag and total points, indicating decrease in absolute amount of points. However, the points are still higher than the points by other players which marks improvement in the rank. This behavior is completely opposite to what is exhibited by a player in low energy efficient class. The player is neither affected by the time of the day, nor by the external factors (Figure 3.7(b)) showing a dedicated effort to save energy. The game engagement behavior for this player, given in Figure 3.7(c) is inconclusive, possibly due to dominance by other energy saving factors.

Test whether X causes Y	Fan ⇒ Ceiling Light		Humidity ⇒ Fan		Desk Light ⇒ Fan		Ceiling Light ⇒ Desk Light		Morning ⇒ Desk Light		Afternoon ⇒ Fan		Evening ⇒ Ceiling Light	
Player type	p-value	F-statistic	p-value	F-statistic	p-value	F-statistic	p-value	F-statistic	p-value	F-statistic	p-value	F-statistic	p-value	F-statistic
Low Energy Efficient	0.54	0.37	**0.004**	8.12	0.06	3.55	0.81	0.06	0.4	0.71	**0.01**	6.1	**0**	25.3
Medium Energy Efficient	**0**	21.2	**0.008**	7.06	**0**	113.6	**0**	25.8	0.23	1.41	0.46	0.55	**0.0007**	11.5
High Energy Efficient	**0**	21.9	0.12	2.36	0.99	0.003	0.93	0.007	0.63	0.22	**0.04**	4.2	0.52	0.41

Table 3.2: Causality test results among various potential causal relationships. In bold are the p-values (shaded in blue) where Granger's causality is established through F-statistic test between features at the 5% significance level.

Causal Relationship between features

To ensure the correctness of results in Section 3.5 and to enhance the explainable nature of our model, we studied the causal relationship between features using Granger's causality test. Granger's causality is a statistical test used to determine causal relationship between two signals. If signal A Granger's-causes signal B, then past values of A can be used to predict B for future timesteps beyond what is available for B. The results for causal relationship study is given in Table 3.2. Under null hypothesis H_0, X does not Granger's-cause Y. So, a p-value lower than 0.05 (5% significance level) indicates a strong causal relationship between the tested features and implies rejecting the null hypothesis H_0.

The p-values (shaded in blue) for which Granger's causality is established are highlighted in the table. Interestingly, for medium and high energy efficient building occupants, ceiling fan usage causes ceiling light usage. This in fact confirms the predictive behavior for them

as mentioned earlier. In both low and medium energy efficient building occupants, external humidity causes ceiling fan usage. This is an indicator that their energy usage is affected by external weather conditions. However, for high energy efficient building occupants external humidity doesn't cause ceiling fan usage. This shows that they are highly engaged with the proposed gamification interface and try to minimize their energy usage. Another interesting result is that the evening label causes ceiling light usage for both low and medium energy efficient building occupants. But this is not the case for high energy efficient building occupants, for whom ceiling light usage is better optimized as a result of their strong engagement with the ongoing social game, eventually leading to exhibition of better energy efficiency.

Labeling unsupervised clusters using feature correlation knowledge from supervised classification

We also learn the feature correlations in clusters obtained from unsupervised clustering of data in Section 3.3. Based on the feature correlation knowledge gained from different supervised classes in Section 3.5, we label the clusters as having low, medium or high energy efficient data. As an illustration, the feature correlations for C_{unsup}^3 is shown in Fig 3.8. It is evident from Fig 3.8(a) that data in C_{unsup}^3 exhibit predictability in behavior with correlations between resource usage flags. Also the weekdays are marked by energy savings in terms of decrease in fan usage minutes. The time of the day is also unrelated to the performance. Neither do the external factors contribute to the performance (Figure 3.8(b)). The engagement in the game also boosts the points (Figure 3.8(c)). All the above behaviors are indicative of the similarity between the energy efficiency characteristics manifested by C_{unsup}^3 and the high energy efficient class obtained using supervised segregation (C_{sup}^{High}). So, C_{unsup}^3 is labeled as the high energy efficient cluster. Following the same comparison, C_{unsup}^1 and C_{unsup}^2 are labeled as the medium and low energy efficient clusters respectively.

To further strengthen our inference, we compute the similarity using Pearson Correlation and RV coefficient [117] between the feature correlation matrices in unsupervised clusters and supervised classes. Figure 3.9 showing the result of above operation confirms our earlier assignment of labels to the unsupervised clusters, i.e. $\{C_{unsup}^1 \sim$ Medium Energy Efficient$\}$,$\{C_{unsup}^2 \sim$ Low Energy Efficient$\}$ and $\{C_{unsup}^3 \sim$ High Energy Efficient$\}$. The labeled unsupervised clusters are the final segments that can be used for a number of downstream tasks as discussed in Section 3.6.

3.6   Conclusion and Future Work

A general framework for segmentation analysis in energy game-theoretic frameworks was presented in this research work. The analysis included clustering of agent behaviors and an explainable statistical model representing the contributed features motivating their decision-making. To strengthen our results, we examined several feature correlations using granger
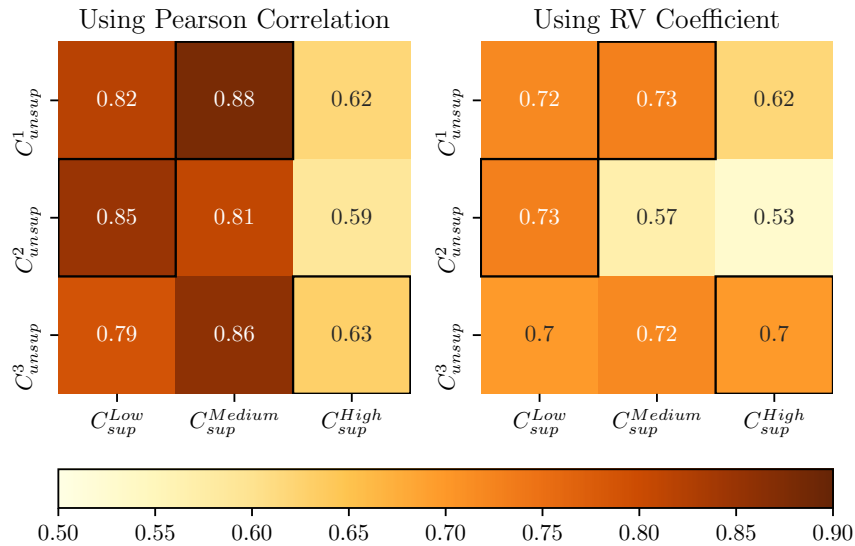
Figure 3.9: Similarity between feature correlation matrices. The highest value in each column is highlighted and corresponds to the matching of supervised classes to the unsupervised clusters

causality test for potential causal relationships. Coupled with statistical justification and explainability, the proposed method can provide characteristic clusters demonstrating different energy use behaviors, following which, specific incentives can be designed for each cluster.

There are several directions for future research. Our ultimate goal for the segmentation analysis is to improve the gamification methodology, to simultaneously learn occupant preferences while also opening avenues for feedback, as static programs for encouraging energy efficiency are less efficient with passing of time [118]. So, an improved version of energy social game, similar in structure to that of [93] but with intelligent incentive design and privacy preserving techniques [119] can be implemented, with building occupants and managers interaction modeled as a reverse stackelberg game (leader-follower) in which there are multiple followers that play in a non-cooperative game setting [111]. By leveraging proposed segmentation analysis, an adaptive model can be formulated that learns how user preferences change over time, and thus generate the appropriate incentives. Furthermore, the learned preferences can be adjusted through incentive mechanisms [120] and a tailored mean-field game approach [121] to enact improved energy efficiency. Above two operations can be carried out in a tree structure, with segmentation carried out in regular intervals in each of the tree branches, as depicted in Figure 3.10. The above can be coherently designed with other smart building systems [8, 45, 53, 122]. Summing up, this would result in a novel mechanism design, effectively enabling variation in occupant's behaviors, in order to meet, for instance, the requirements of a demand response program. Another line of future work can be to study the delayed impacts of energy social game and design it accordingly to achieve long term energy efficiency, like a research in same line [123].
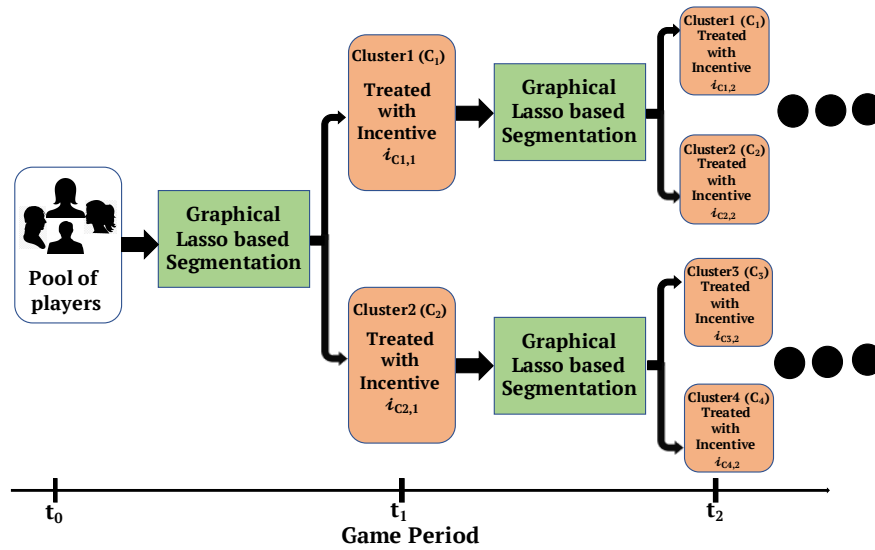
Figure 3.10: Tree based incentive design mechanism employing proposed graphical lasso based segmentation method. Clusters are treated with incentives specifically tailored for them.

Chapter 4

Likelihood Contribution based Multi-scale Architecture for Generative Flows

4.1 Introduction

Deep Generative Modeling aims to learn the embedded distributions and representations in input (especially unlabeled) data, requiring no/minimal human labeling effort. The representations learnt can then be utilized in a number of downstream tasks such as semi-supervised learning [124, 125], synthetic data augmentation and adversarial training [126], text analysis and model based control etc. The repository of deep generative modeling majorly includes likelihood based models such as autoregressive models, latent variable models, flow based models and implicit models such as generative adversarial networks (GANs). Autoregressive models [127–130] achieve exceptional log-likelihood score on many standard datasets, indicative of their power to model the inherent distribution. But, they suffer from a slow sampling process. Latent variable models such as variational autoencoders [131] tend to better capture the global feature representation in data, but do not offer an exact density estimate. Implicit generative models such as GANs optimize a generator and a discriminator in a min-max fashion have recently become popular for their ability to synthesize realistic data [132, 133]. But, GANs neither offer a latent space suitable for further downstream tasks, nor do they perform density estimation.

Flow based generative models [1, 134] perform exact density estimation with fast inference and sampling, due to their parallelizability. They also provide an information rich latent space suitable for many applications. However, the dimension of latent space for flow based generative models is same as the high-dimensional input space, by virtue of the bijective nature of flows. This poses a bottleneck for flow models to scale with increasing input dimensions due to the computational complexity. An effective solution to the above problem is a multi-scale architecture, introduced by [1], which performs iterative early gaussianization

of a part of the total dimensions at regular intervals of flow layers. This not only makes the model computational and memory efficient but also aids in distributing the loss function throughout the network for better training. Many prior works including [134–137] implement multi-scale architecture in their flow models, but use static masking methods for factorization of dimensions.

We propose a multi-scale architecture which performs data dependent factorization to decide which dimensions should pass through more flow layers. For the decision making, we introduce a heuristic based on the amount of log-likelihood contribution by each dimension, which in turn signifies their individual importance. Since in the proposed architecture, the heuristic is obtained as part of the flow training process, it can be universally applied to generic flow models. We present such implementations for flow models based on affine/additive coupling as well as ordinary differential equations (ODE) and achieve quantitative and qualitative improvements. We also perform ablation studies to establish the novelty of our method [138, 139]. Summing up, the contributions of our research:

1. A log-determinant based heuristic which entails the contribution by each dimension towards the total log-likelihood in a multi-scale architecture.

2. A multi-scale architecture based on the above heuristic performing data-dependent splitting of dimensions, implemented for several classes of flow models.

3. Quantitative and qualitative analysis of above implementations and an ablation study

To the best of our knowledge, we are the first to propose a data-dependent splitting of dimensions in a multi-scale architecture.

4.2   Background

Flow-based Generative Models

Let $\mathbf{x}$ be a high-dimensional random vector with unknown true distribution $p(\mathbf{x})$. The following formulation is directly applicable to continous data, and with some pre-processing steps such as dequantization [127, 140, 141] to discrete data. Let $\mathbf{z}$ be the latent variable with a known standard distribution $p(\mathbf{z})$, such as a standard multivariate gaussian. Using an i.i.d. dataset $\mathcal{D}$, the target is to model $p_{\boldsymbol{\theta}}(\mathbf{x})$ with parameters $\boldsymbol{\theta}$. A flow, $\mathbf{f}_{\boldsymbol{\theta}}$ is defined to be an invertible transformation that maps observed data $\mathbf{x}$ to the latent variable $\mathbf{z}$. A flow is invertible, so the inverse function $\mathcal{T}$ maps $\mathbf{z}$ to $\mathbf{x}$, i.e.

$$\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathcal{T}^{-1}(\mathbf{x}) \ \ \text{and} \ \ \mathbf{x} = \mathcal{T}(\mathbf{z}) = \mathbf{f}_{\boldsymbol{\theta}}^{-1}(\mathbf{z}) \tag{4.1}$$

The log-likelihood can be expressed as,

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log p(\mathbf{z}) + \log \left| \det \left(\frac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}^T} \right) \right| \tag{4.2}$$

where $\dfrac{\partial \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}^T}$ is the Jacobian of $\mathbf{f}_{\boldsymbol{\theta}}$ at $\mathbf{x}$. The invertible nature of a flow allows it to be capable of being composed of other flows of compatible dimensions. In practice, flows are constructed by composing a series of component flows. Let the flow $\mathbf{f}_{\boldsymbol{\theta}}$ be composed of K component flows, i.e. $\mathbf{f}_{\boldsymbol{\theta}} = \mathbf{f}_{\theta_K} \circ \mathbf{f}_{\theta_{K-1}} \circ \cdots \circ \mathbf{f}_{\theta_1}$ and the intermediate variables be denoted by $\mathbf{y}_K, \mathbf{y}_{K-1}, \cdots, \mathbf{y}_0 = \mathbf{x}$. Then the log-likelihood of the composed flow is,

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \underbrace{\log p(\mathbf{z})}_{\text{Log-latent density}} + \underbrace{\log \left| \det \left(\frac{\partial (\mathbf{f}_{\theta_K} \circ \mathbf{f}_{\theta_{K-1}} \circ \cdots \circ \mathbf{f}_{\theta_1}(\mathbf{x}))}{\partial \mathbf{x}^T} \right) \right|}_{= \sum_{i=1}^{K} \log |\det(\partial \mathbf{y}_i / \partial \mathbf{y}_{i-1}^T)| \ \text{(Log-det)}} \tag{4.3}$$

which follows from the fact that $\det(A \cdot B) = \det(A) \cdot \det(B)$. In our work, we refer the first term in Equation 4.3 as *log-latent-density* and the second term as *log-determinant (log-det)*. The reverse path, from $\mathbf{z}$ to $\mathbf{x}$ can be written as a composition of inverse flows, $\mathbf{x} = \mathbf{f}_{\boldsymbol{\theta}}^{-1}(\mathbf{z}) = \mathbf{f}_{\theta_1}^{-1} \circ \mathbf{f}_{\theta_2}^{-1} \circ \cdots \circ \mathbf{f}_{\theta_K}^{-1}(\mathbf{z})$. Confirming with above properties for a flow, different types of flows can be constructed [1, 134, 135, 142, 143].

Multi-scale Architecture

Multi-scale architecture is a design choice for latent space dimensionality reduction of flow models, in which part of the dimensions are factored out/early gaussianized at regular intervals, and the other part is exposed to more flow layers. The process is called dimension factorization. In the problem setting as introduced in Section 4.2, the factoring operation can be mathematically expressed as,

$$\mathbf{y}_0 = \mathbf{x}, (\mathbf{z}_{l+1}, \mathbf{y}_{l+1}) = \mathbf{f}_{\theta_{l+1}}(\mathbf{y}_l), \quad l \in \{0, 1, \cdots, K-2\}$$
$$\mathbf{z}_K = \mathbf{f}_{\theta_K}(\mathbf{y}_{K-1}), \mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_K)$$

The factoring of dimensions at early layers has the benefit of distributing the loss function throughout the network [1] and optimizing the amount of computation used by the model.

4.3 Likelihood Contribution based Multi-scale Architecture

In a multi-scale architecture, it is apparent that the network will better learn the distribution of dimensions getting exposed to more layers of flow as compared to the ones which get factored at a finer scale (earlier layer). The method of dimension splitting proposed by prior works such as [1, 134, 135] are static in nature and do not distinguish between importance of different dimensions. In this section, we introduce the general framework for likelihood contribution based heuristic and associated multi-scale architecture along with its integration with flow training process.

Likelihood Contribution based Heuristic

Recall from Equation 4.3 that the log-likelihood is composed of two terms, the log-latent density term and the log-det term. We focus on the log-det term since it depends on the modeling of flow layers.

Let the dimension of the input (images in our case) space $\mathbf{x}$ be $s \times s \times c$, where s is the height/width and c is the number of channels. For the following formulation, let us assume NO dimensions were gaussianized early so that we have access to log-det term for all dimensions at each flow layer, and the dimension at all intermediate layers remain the same (i.e. $s \times s \times c$). We apply a flow ($\mathbf{f_\theta}$) with K component flows to $\mathbf{x}$, $\mathbf{z}$ pair, so that $\mathbf{z} = \mathbf{f_\theta}(\mathbf{x}) = \mathbf{f}_{\theta_K} \circ \mathbf{f}_{\theta_{K-1}} \circ \cdots \circ \mathbf{f}_{\theta_1}(\mathbf{x})$. The intermediate variables are denoted by $\mathbf{y}_K, \mathbf{y}_{K-1}, \cdots, \mathbf{y}_0$ with $\mathbf{y}_K = \mathbf{z}$ (since no early gaussianization was performed) and $\mathbf{y}_0 = \mathbf{x}$. The log-det term at layer l, $L_d^{(l)}$, is given by,

$$[L_d^{(l)}]_{scaler} = \sum_{i=1}^{l} \log|\det(\partial \mathbf{y}_i / \partial \mathbf{y}_{i-1}^T)| \tag{4.4}$$

The log-det of the jacobian term encompasses contribution by all the $s \times s \times c$ dimensions. If we decompose it to obtain the individual contribution by the dimensions (we discuss explicitly on how to perform this decomposition in Sec. 4.3) towards the total log-det ($\sim$ total log-likelihood). The log-det term can be viewed (with slight abuse of notations) as a $s \times s \times c$ tensor corresponding to each of the dimensions, summed over the flow layers till l,

$$[L_d^{(l)}]_{s\times s\times c} = \sum_{i=1}^{l} [d_{i-1}^{(\alpha,\beta,\gamma)}]_{s\times s\times c},$$

where $\alpha, \beta \in \{0, \cdots, s\}$ and $\gamma \in \{0, \cdots, c\}$, s.t. $\sum_{\alpha,\beta,\gamma} d_{i-1}^{(\alpha,\beta,\gamma)} = \log|\det(\partial \mathbf{y}_i / \partial \mathbf{y}_{i-1}^T)|$

The entries in $[L_d^{(l)}]_{s\times s\times c}$ having higher value were scaled up more, and correspond to dimensions which are more sensitive to changes in input, so the flow can learn more by processing them through more layers. So, we can use the *likelihood contribution (in the form of log-det term) by each dimension* as a heuristic for deciding which variables should be gaussianized early in a multi-scale architecture.

Estimation of Per-Dimension Likelihood Contribution for various Flow types

The likelihood (log-det) contributions per dimension can be obtained by decomposition of the overall log-det of the jacobian. Now, we describe the decomposition process for various types of flow models. The log-det per dimension after decomposition is averaged across the samples in the training set, so as to obtain an overall representative likelihood contribution by each dimension.

Affine coupling based flows

RealNVP [1]: For RealNVP with affine coupling layers, the logarithm of individual diagonal elements of jacobian, summed over layers till l provides the per-dimensional likelihood contribution components at layer l.

Glow [134]: Unlike RealNVP where the log-det terms for each dimension can be expressed as log of corresponding diagonal element of jacobian, Glow contains 1×1 convolution blocks having non-diagonal log-det term. For a $s \times s \times c$ tensor, the log-det term is $s \cdot s \cdot \log |\det(W)|$, where W is the weight matrix. It is clear that the contribution by a pixel is $\log |\det(W)|$, and it has to be decomposed to obtain individual contribution by each channel. As a suitable candidate, singular values of W correspond to the contribution from each channel dimension, hence their log value is the individual log-det contribution. The individual log-det term for channels are obtained by,

$$|\det(W)| = \prod_i \sigma_i(W) \Leftrightarrow \log |\det(W)| = \sum_i \log(\sigma_i(W)) \tag{4.5}$$

where $\sigma_i(W)$ are the singular values of W. For affine blocks, same method as RealNVP is adopted.

Flow models with ODE based Density Estimators

Recent works on flow models such as [135, 143, 144] employ variants of ODE based density estimators. The following formulation is applicable to find per-dimensional likelihood contributions for such flow models. In the above works, the flow is modelled as $F(x)$, such that $z = F(x) = (I + g)(x)$, where $g(x)$ is the forward propagation function. The log-det of the jacobian is expressed as a power series,

$$\ln |\det J_F(x)| = \mathrm{tr}\left(\ln\left(I + J_g(x)\right)\right) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{\mathrm{tr}(J_g^k)}{k}$$

where tr denotes the trace. Due to computational constraints, the power series is computed up to a finite number of iterations with the $\mathrm{tr}(J_g^k)$ term stochastically approximated by hutchinson's trace estimator, $\mathrm{tr}(A) = \mathbb{E}_{p(v)}\left[v^T A v\right]$, with $\mathbb{E}[v] = 0$ and $\mathrm{Cov}(v) = I$. The component corresponding to each dimension that becomes part of the log-det term is the diagonal element of J_g^k, so the per-dimension contribution to the likelihood can be approximated as the diagonal elements of J_g^k, summed over the power series upto a finite number of iterations n. The diagonal elements are obtained with the hutchinson's trace estimator without any extra cost, i.e. if $v = [v_1, v_2, \cdots, v_{s \times s \times c}]^T$,

$$\sum_{k=1}^{\infty} (-1)^{k+1} \frac{\mathrm{tr}(J_g^k)}{k} = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{\mathbb{E}_{p(v)}\left[v^T J_g^k v\right]}{k} = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{\mathbb{E}_{p(v)}\left[(v^T J_g^k)v\right]}{k}$$

Algorithm 1: LCMA Implementation and Training for Generative Flow models

Pre-training Phase: Pre-train a flow model with no multi-scale architecture (no dimensionality reduction) to obtain the log-det terms ($[L_d^{(l)}]_{s \times s \times c}$) at each layer l

Dimension Factorization Phase: Initialize the input dimensions $\phi \times \phi \times \psi \leftarrow s \times s \times c$

while $1 \leq l \leq K$ **do**

$\quad$ Select $[L_d^{(l)}]_{\phi \times \phi \times \psi}$ corresponding to input dimensions.

$\quad [L_d^{(l)}]_{\phi \times \phi \times \psi} \xrightarrow[\text{(Figure 1)}]{\text{Local Max- \& Min-Pooling}} [L_d^{(l)}]_{\frac{\phi}{2} \times \frac{\phi}{2} \times 4\psi} \xrightarrow[\text{Splitting}]{\text{Channelwise}} [L_d^{(l,\text{Max})}, L_d^{(l,\text{Min})}]_{\frac{\phi}{2} \times \frac{\phi}{2} \times 2\psi}$

$\quad$ Gaussianize the dimensions corresponding to $L_d^{(l,\text{Min})}$ and pass the dimensions corresponding to $L_d^{(l,\text{Max})}$ to more flow layers

$\quad \phi \times \phi \times \psi \leftarrow \frac{\phi}{2} \times \frac{\phi}{2} \times 2\psi$

end

Training Phase: Flow model with proposed LCMA is trained using maximum likelihood.

In the above equation, $(v^T J_g^k)$ is the vector-jacobian product which is multiplied again with v. The individual components which are summed when $(v^T J_g^k)$ is multiplied with v correspond to the diagonal terms in jacobian, over the expectation $\mathbb{E}_{p(v)}$. So those terms are the contribution by the individual dimensions to the log-likelihood and are expressed as $[L_d^{(l)}]_{s \times s \times c}$ at flow layer l.

Dimension Factorization using Proposed Heuristic

At every flow layer, an ideal factorization method should,

1. *(Quantitative) For efficient density estimation:* Early gaussianize the dimensions having less log-det and expose the ones having more log-det to more flow layers. In this manner, selectively enhanced expressivity can be provided to dimensions which capture meaningful representations (and are more valuable from a log-det perspective).

2. *(Qualitative) For qualitative reconstruction:* Capture the local variance over the flow layers, i.e. the dimensions being exposed to more flow layers should contain representative pixel variables from throughout the whole image.

We perform a hybrid dimension factorization taking both of the above criterias into account. A $s \times s \times c$ tensor is converted to $\frac{s}{2} \times \frac{s}{2} \times 4c$ tensor, by using local max and min pooling operations on corresponding per dimensional log-det terms as obtained in Sec. 4.3 (which was averaged over the training set, so the learned splitting remains same for all data points). Then the $\frac{s}{2} \times \frac{s}{2} \times 4c$ tensor is split along channel dimension to form two $\frac{s}{2} \times \frac{s}{2} \times 2c$ tensors, one corresponding to low log-det dimensions and one corresponding to high ones.
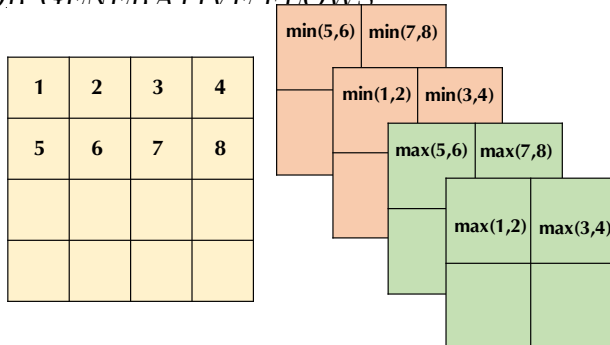
Figure 4.1: Likelihood contribution based squeezing operation: (On left) The tensor $[L_d^{(l)}]_{s \times s \times c}$ representing log-det of variables in a flow layer. (On right) It is squeezed to $\frac{s}{2} \times \frac{s}{2} \times 4c$ with local max and min pooling operation. The green (orange) marked pixels represent dimensions having more (less) log-det locally.

The local min and max pooling operations (illustrated in Fig 4.1) preserve the local spatial variation of the image in both parts of the factorization, leveraging both enhanced density estimation as well as qualitative reconstruction.

Factorization for Flows involving Squeezing Operation: If squeezing operation (which is nothing but reordering of dimensions) is involved, we keep track of which dimensions belong to the half that get gaussianized early and which dimensions belong to the other half that passes through more flow layers. At the next layer, only the log-det terms for the dimensions which came through flow layers are considered for further splitting operation.

We refer to the multi-scale architecture obtained by the above data-dependent dimension splitting method as Likelihood Contribution based Multi-scale Architecture (LCMA).

Integration of LCMA with Flow Model Training

Training of flow models with LCMA implementation is summarized in Algorithm 1.

Pre-training Phase involves training a non-multi-scale (NMS) flow model to obtain the individual contribution of dimensions ($[L_d^{(l)}]_{s \times s \times c}$) towards the total log-likelihood (Section 4.3). Training a NMS model is computation heavy, but better training leads to improved density estimation score for resulting LCMA. So, there remains a trade-off between the amount of NMS pre-training and the density estimation performance for resulting LCMA. We train the NMS model partially (we report the results for varying levels of NMS pre-training and resulting density estimation performance for one of the flow training in the supplementary materials).

Dimension Factorization Phase involves deciding the dimensions to be factored out at each flow layer using proposed log-det heuristic. The dimension splittings are decided

recursively for all the flow layers based on their likelihood contribution obtained in the previous phase.

Training Phase is the final stage where the flow model with LCMA is trained. Since the decision for factorization of dimensions at each flow layer occurs before the training starts, and the decision remains unchanged during training, change of variables formula can be applied. In fact, this allows the use of non-invertible operations (e.g. max and min pooling) for efficient factorization (Sec. 4.3).

Time Complexity: The time complexity associated with original multi-scale architecture is not affected by LCMA implementation as no additional time consuming blocks were added. Our data-dependent dimension splitting operation can be interpreted as replacing the conventional checkerboard/channel split masking with likelihood contribution based masking.

4.4 Related Work

For multi-scale architectures in generative flow models, our proposed method performs factorization of dimensions based on their likelihood contribution, which in another sense translates to determining which dimensions are important from density estimation and qualitative reconstruction point of view. Keeping this in mind, we discuss prior works on generative flow models which involve multi-scaling and/or incorporate permutation among dimensions to capture their interactions.

A number of generative flow models implement a multi-scale architecture, such as [1, 134–136, 143, 145, 146] etc. [134] introduce an 1×1 convolution layer in between the actnorm and affine coupling layer in their flow architecture. The 1×1 convolution is a generalization of permutation operation which ensures that each dimension can affect every other dimension. This can be interpreted as redistributing the contribution of dimensions to total likelihood among the whole space of dimensions. So [134] treat the dimensions as equiprobable for factorization in their implementation of multi-scale architecture, and split the tensor at each flow layer evenly along the channel dimension. We, on the other hand, take the next step and focus on the individuality of dimensions and their importance from the amount they contribute towards the total log-likelihood. The log-det score is available via direct/indirect decomposition of the jacobian obtained as part of computations in a flow training, so we essentially have an easily available heuristic. Since LCMA focuses individually on the dimensions using easily available heuristic, it can prove to be versatile in being compatible with generic multi-scale architectures. [147] extend the concept of 1×1 convolutions to invertible $d \times d$ convolutions, but do not discuss multi-scaling. [1] also mention a type of permutation which is equivalent to reversing the ordering of the channels, but is more restrictive and fixed.

Flow models such as [135, 143, 144] involve ODE based density estimators. They also implement a multi-scale architecture, but the splitting operation is a static channel wise splitting without considering the importance of individual dimensions or any permutations. [136, 137, 145, 146] use multi-scale architecture in their flow models, coherent with [1, 134], but still perform the factorization of dimensions using static masking. For qualitative

Table 4.1: Improvements in density estimation (in bits/dim) using proposed method for RealNVP

Model	CelebA	CIFAR-10	ImageNet 32x32	ImageNet 64x64
RealNVP	3.02	3.49	4.28	3.98
RealNVP with LCMA	**2.71**	**3.43**	**4.21**	**3.92**

sampling along with efficient density estimation, we also propose that factorization methods should preserve spatiality of the image in the two splits, motivated by the spatial nature of splitting methods in [134] (channel-wise splitting) and [1] (checkerboard and channel-wise splitting). Summarizing, we propose a data-dependent approach to dimension factorization in a multi-scale architecture, unexplored by prior works.

4.5 Experiments

In this section we present the detailed results of proposed LCMA adopted for the flow model of RealNVP [1], Glow [134],i-ResNet [135] and Residual Flows [143]. For direct comparison, all the experimental settings such as data pre-processing, optimizer parameters as well as flow architectural details (coupling layers, residual blocks) are kept the same, only the factorization of dimensions at each flow layer is performed as per LCMA. The computations were performed in NVIDIA Tesla V100 GPUs. For RealNVP, we perform experiments on four benchmarked image datasets: *CIFAR-10* [148], *Imagenet* [149] (downsampled to 32×32 and 64×64), and *CelebFaces Attributes (CelebA)* [150]. The scaling in LCMA is performed once for CIFAR-10, thrice for Imagenet 32×32 and 4 times for Imagenet 64×64 and CelebA. We compare LCMA with conventional RealNVP and report the quantitative and qualitative results. For Glow, i-ResNet and Residual Flows with LCMA, we perform experiments on CIFAR-10 and report improvements over baseline bits/dim (BPD).

Quantitative Comparison

The bits/dim scores of RealNVP with conventional multi-scale architecture and RealNVP with LCMA are given in Table 4.1. It can be observed that the density estimation results using LCMA is in all cases better in comparison to the baseline. We observed that the improvement for CelebA is relatively high as compared to natural image datasets. This observation was expected as facial features often contain high redundancy and the flow model learns to put more importance (reflected in terms of high log-det) on selected dimensions that define the facial features. Proposed LCMA exposes such dimensions to more flow layers, making them more expressive and hence the significant improvement in BPD is observed. The improvement in bits/dim is less for natural image datasets because of the high variance among

Table 4.2: Density estimation results (in bits/dim) for various flow models with LCMA on CIFAR-10

Type of Multi -scale Architecture (MA)	RealNVP	Glow	i-ResNet	Residual Flows
Conventional MA	3.49	3.35	3.45	3.28
LCMA	**3.43**	**3.31**	**3.40**	**3.25**

features defining them, which has been the challenge with image compression algorithms. Note that the improvement in density estimation is always relative to the original flow architecture (RealNVP in our case) over which we use our proposed LCMA, as we do not alter any architecture other than the dimension factorization method.

The quantitative results of LCMA implementation for several state-of-the-art flow models with CIFAR-10 dataset is summarized in Table 4.2. The density estimation score for flow with LCMA outperforms the same flow with conventional multi-scale architecture. We also achieve state-of-the-art density estimation results for CIFAR-10 dataset with LCMA implementation for Residual Flows.

Qualitative Comparison

For LCMA implementation, we introduced local max and min pooling operations (to preserve spatiality) on log-det heuristic to decide which dimensions to be gaussianized early (Section 4.3). Fig. 4.2(a) shows samples from original datasets, Fig. 4.2(b) shows the samples from a trained RealNVP flow model with conventional multi-scale architecture and Fig. 4.2(c) shows the samples from RealNVP with LCMA, trained on various datasets. The finer facial details such as hair styles, eye-lining and facial folds in Celeba samples generated from RealNVP with LCMA were perceptually better than the baseline. The global feature representation observed is similar to that in RealNVP, as the flow architecture was kept the same. The background for natural images such as Imagenet were constructed at par with the original flow model. As it has been observed, for flow models, the latent space holds knowledge about the feature representation in the data. We performed linear interpolations in latent space to ensure its efficient construction. The interpolations observed (Fig. 4.3) were smooth, with intermediate samples perceptibly resembling synthetic faces, signifying the efficient construction of latent space.

Ablation Study

We perform two types of ablation studies to compare LCMA with other methods for dimension factorization in a multi-scale architecture. In our first study, we consider 4 variants, namely fixed random permutation (Case 1), multiscale architecture with early gaussianization of high

(a) Examples from the dataset   (b) Samples from trained Real-   (c) Samples from trained Real-
                                 NVP [1]   NVP flow model with LCMA

Figure 4.2: Samples from RealNVP [1] and RealNVP flow model with proposed LCMA trained on different datasets. The datasets shown in this figure are in order: CIFAR-10, Imagenet(32×32), Imagenet (64×64) and CelebA (without low-temperature sampling).

Figure 4.3: Smooth linear interpolations in latent space between two images from CelebA. The intermediate samples perceptibly resemble synthetic faces.

log-det dimensions (Case 2), factorization method with checker-board and channel splitting

Table 4.3: Ablation study results for multi-scale architectures with various factorization methods trained on CelebA dataset

Evaluations	Fixed Random Permutation	Early gaussianization of *high* log-det dimensions	RealNVP	Early gaussianization of *low* log-det dimensions
Quantitative (BPD)	3.05	3.10	3.02	2.71
Qualitative				

as introduced in RealNVP (Case 3) and multiscale architecture with early gaussianization of low log-det dimensions, which is our proposed LCMA (Case 4). In fixed random permutation, we randomly partition the tensor into two halves, with no regard to the spatiality or log-det score. In case 2, we do the reverse of LCMA, and early gaussianize the high log-det variables at each layer. The bits/dim score and generated samples for each of the methods are given in Table 4.3. As expected from an information theoretic perspective, gaussianizing high log-det variables early provides the worst density estimation, as the model could not capture the high amount of important information. Comparing the same with fixed random permutation, the latter has better score as the probability of a high log-det variable being gaussianized early reduces to half, and it gets further reduced with RealNVP due to channel-wise and checkerboard splitting. LCMA has the best score among all methods, as the variables more sensitive to changes in input (hence carrying more information) are exposed to more flow layers. Fixed random permutation has the worst quality of sampled images, as the spatiality is lost during factorization. The sample quality improves for Case 2 and RealNVP. The sampled images are perceptually best for LCMA.

We perform a second ablation study to reconfirm that early gaussianization of high log-det dimensions has a deteriorating effect on the density estimation score. The flow model in our experiment has 3 layers where dimensions splitting is being performed. We consider all permutations of early gaussianization of high/low log-det dimensions at each of the 3 layers. The density estimation scores for all 2^3 permutations trained on CelebA dataset are presented in Table 4.4. The best score corresponds to early gaussianization of low log-det dimensions at each flow layer (proposed LCMA), and the score deteriorates with permutations involving early gaussianization of high log-det dimensions at any flow layer. Summarizing, LCMA outperforms multi-scale architectures based on other factorization methods, as it improves density estimation scores and generates qualitative samples.

Table 4.4: Ablation study results for permuting factorization of high/low log-det dims

Permutation of high/low log-det dimensions	Bits/dim
High-High-High	3.10
High-High-Low	3.09
High-Low-High	3.07
High-Low-Low	3.05
Low-High-High	3.00
Low-High-Low	2.92
Low-Low-High	2.79
Low-Low-Low (LCMA)	2.71

4.6   Conclusions

We propose a novel multi-scale architecture for generative flows which employs a data-dependent splitting based on the individual contribution of dimensions to the log-likelihood. Implementations of the proposed method for several flow models such as RealNVP [1], Glow [134], i-ResNet [135] and Residual Flows [143] were presented. Empirical studies conducted on benchmark image datasets validate the strength of our proposed method, which improves log-likelihood scores and is able to generate qualitative samples. Ablation study results confirm the power of LCMA over other options for dimension factorization. A line of future work can be to design/learn a masking scheme for factorization online during flow training (or possibly a parallel training process), while preserving flow properties.

Part II

Conditional Synthetic Data Generation

Chapter 5

Conditional Synthetic Data Generation

5.1   Introduction

In Section 1.2, we covered the data scarcity and class-imbalance challenge observed at the onset of a pandemic, since the availability of data corresponding to the new disease is scarce. In this chapter, we present a novel conditional synthetic data-generation method to augment the available pandemic data of interest. Our proposed method can also help organizations release synthetic versions of their actual data with similar behavior in a privacy-preserving manner. At the onset of a pandemic, when the availability of disease data is limited, our proposed model learns the distribution of available limited data and then generates conditional synthetic data that can be added to the existing data in order to improve the performance of machine learning algorithms. To tackle the challenge of label scarcity, we propose semi-supervised learning methods to leverage the small amount of labeled data and still generate qualitative synthetic samples. Our methods can enable healthcare ML tools to rapidly adapt to a pandemic.

We apply this method to generate conditional CT scan images corresponding to COVID cases (Fig. 5.1), and conduct qualitative and quantitative tests to ensure that our model generates high-fidelity samples and is able to preserve the features corresponding to the condition (COVID/Non-COVID) in synthetic samples. As a downstream use of conditional synthetic data, we improve the performance of COVID-19 detectors based on CT scan data
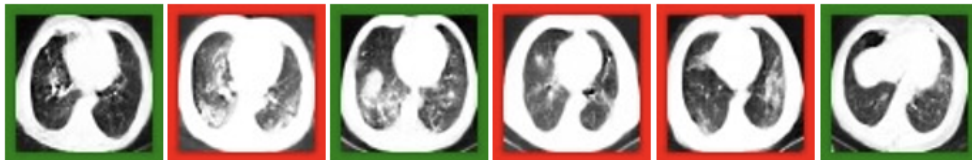
Figure 5.1: Synthetic CT scans generated by our proposed model, with Non-COVID (normal and pneumonia cases, with green border)/ COVID (with red border) as the condition.
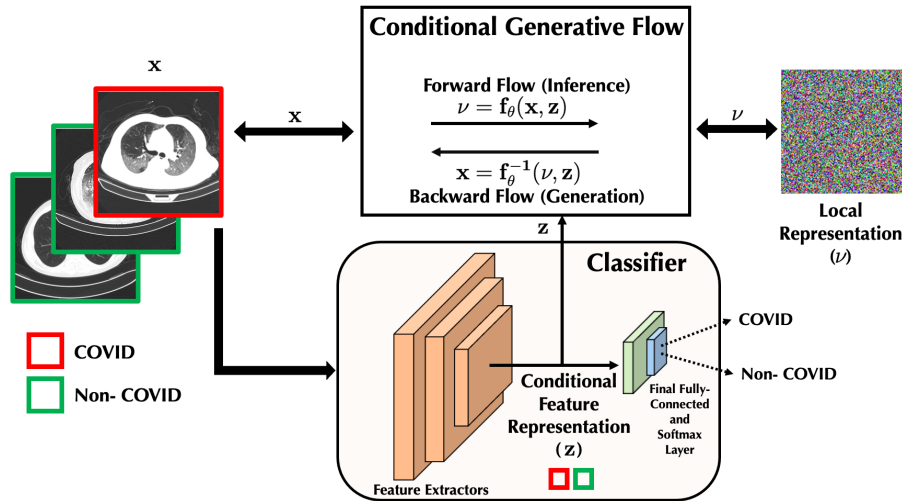
Figure 5.2: Illustration of the proposed conditional synthetic generation. (Best viewed in color)

via synthetic data augmentation. Our results show that the proposed model is able to generate synthetic data that mimic the real data, and the generated samples can indeed be augmented with existing data in order to improve COVID-19 detection efficiency.

The proposed methods are developed such that they are generalizable across applications. To illustrate this, we implement the above algorithm to generate synthetic personal thermal comfort data, based on the data collected in [8], and described in Section 2.2.

5.2   Methodology

We present a hybrid model consisting of a conditional generative flow and a classifier for conditional synthetic generation. We also introduce a semi-supervised approach, to generate conditional synthetic samples when a few samples out of the whole dataset are labeled.

COVID and Non-COVID Classifier

Our model is characterized by the efficient decoupling of feature representations corresponding to the condition and the local noise. Suppose we have N samples $\mathbf{X}$ with labels Y, with 2 possible classes, COVID/Non-COVID. We first train a classifier C (consisting of a feature extractor network denoted by $g(\cdot)$, and a final fully-connected and softmax layer, denoted by $h(\cdot)$, i.e. $C(x) = h(g(x))$) to classify the input sample (which in our case are CT Scans) and associated labels as COVID and Non-COVID. Mathematically, this step solves the following

minimization with backpropagation:

$$\min_C \mathcal{L}_C(\mathbf{X}, Y) = -\mathbb{E}_{(x,y)\sim(\mathbf{X},Y)} \sum_{l=1}^{2} \left[\mathbb{I}_{[l=y]} \log C(x)) \right] \qquad (5.1)$$

By virtue of the training process, the classifier learns to discard local information and preserve the features necessary for classification (conditional information) towards the downstream layers. Once the classifier is trained, we freeze its parameters, and use it to extract the conditional (COVID/Non-COVID) feature representation $z = g(x)$ (as a vector without spatial characteristics) at the output of the feature extractor network for input image x. The dimension of z is chosen such that $\dim(z) << \dim(x)$.

Conditional Generative Flow

During the training phase for the flow model, the conditional feature representation z is fed to the conditional generative flow. The flow model is trained using maximum-likelihood, transforming x to its local representation ν, i.e.

$$f_\theta(x, z) = \nu \sim \mathcal{N}(0, I) \qquad (5.2)$$

with ν having the same dimension as x by the inherent design of flow models. We use the method introduced by [151] to incorporate the conditional input z in flow model. Coupling layers in affine flow models have scale ($s(\cdot)$) and shift ($b(\cdot)$) networks [139, 152], which are fed with inputs after splitting, and their outputs are concatenated before passing on to the next layer. We incorporate the conditional information z in the scale and shift networks. Mathematically, (with x as the input, D as input dimension, d as the split size, and y as output of the layer),

$$x_{1:d}, x_{d+1:D} = \text{split}(x)$$
$$y_{1:d} = x_{1:d}$$
$$y_{d+1:D} = s(x_{1:d}, z) \odot x_{d+1:D} + b(x_{1:d}, z)$$
$$y = \text{concat}(y_{1:d}, y_{d+1:D})$$

Since flow models are bijective mappings, the exact x can be reconstructed by the inverse flow with z and ν as inputs. During the generation phase, for an input sample x, we compute the conditional feature representation z. Keeping the conditional feature representation the same, we sample a new local representation $\tilde{\nu}$, and generate a conditional synthetic sample $\tilde{x}$, i.e.

$$\tilde{\nu} \in \mathcal{N}(0, I), \quad \tilde{x} = f_\theta^{-1}(\tilde{\nu}, z) \qquad (5.3)$$

Here, $\tilde{x}$ has the same conditional (COVID/Non-COVID) features as x , but has a different local representation. An illustration of the proposed model is provided in Fig. 5.2 and the steps for the inference and generation phases are summarized in Table 5.1.

Semi-supervised Learning for Conditional Synthetic Generation under Label Scarcity

In reality, often a small amount of the already limited pandemic data available are labeled. Consider this case when a few of the datapoints are labeled, denoted by $\{\mathbf{X}^l, Y^l\}$. The rest of the data (unlabeled) is denoted by $\mathbf{X}^u$. To generate conditional synthetic samples under such label scarce situations, we propose a semi-supervised method to modify the classifier design process, in order to effectively decouple the feature representations corresponding to the conditions.

Label learning algorithm

We first design a label learning algorithm to assign presumptive labels $\tilde{Y}^l$ to the unlabeled samples $\mathbf{X}^u$. Assuming k_i labeled samples are available for class i, we train the classifier network using the labeled samples only and compute in the embedded (z) space (1) the centroid vector c_i for each class and (2) a similarity metric between each unlabeled target sample $x^u \in \mathbf{X}^u$ and the specific centroid. Depending on the dimension of the transformed feature space, this similarity metric can simply be a Gaussian kernel to capture local similarity [153], or the inverse of Wasserstein distance [154] for better generalization with complex networks.

Semi-supervised model training

Ideally, the semi-supervised scheme should be able to (1) identify the correct labels of unlabeled target samples, and (2) update the classifier with the additional information. We establish an alternating approach that recursively performs (1) fixing the feature mapping g and propagating presumptive labels using a greedy assignment, i.e., an unlabeled sample is presumed to have the same label to its closest centroid, and (2) updating the feature mapping (the classifier) as supervised learning by treating the presumptive labels as true labels.

Inference Phase	Generation Phase
1. **(Classifier)** Train the COVID and Non-COVID classifier.	1. **(Classifier)** Corresponding to an input sample x, find its conditional feature representation z using the trained classifier.
2. **(Flow)** For each input sample x,	2. **(Flow)** Sample a local representation $\tilde{\nu} \sim \mathcal{N}(0, I)$.

2.1 Feed x to the classifier and extract the conditional feature representation z from its penultimate layer.
    2.2 Get the local representation as $\nu = f_\theta(x, z)$
    2.3 Train the flow model with maximum-likelihood.
	3. **(Flow)** Get a synthetic sample $\tilde{x} = f_\theta^{-1}(\tilde{\nu}, z)$.

Table 5.1: Summary of steps for conditional inference and generation

The proposed greedy propagation, intuitively simple and practically easy to implement, in fact has theoretical guarantees since the entropy objective is approximately submodular when the feature mapping is fixed. Please refer to [155] for a detailed theoretical analysis. The above is conducted alternately until the convergence of the feature mapping and presumptive label assignment. In practice, the convergence is usually achieved in a few iterations. Once the classifier has been trained with this semi-supervised approach, the conditional generative flow training is performed as specified before in conditional generation section.

5.3   Experiments

Data Collection and Pre-processing

We conduct experiments on chest CT scan data based on the COVIDx CT-1 dataset [57], publicly available in Kaggle[1].

CT Scan Data: The dataset consists of 45,758 images corresponding to healthy individuals, 36,856 images corresponding to individuals afflicted with common pneumonia, and 21,395 images corresponding to individuals with COVID-19.

Pre-processing: We combine the images in the Normal and Pneumonia classes into a single Non-COVID class. We use the train, validation, and test splits defined by the official annotation files. In addition to class labels, the annotations include bounding boxes for the lungs region in the whole CT scans image. We crop the images as per the bounding box and resize them to 64×64.

Hyperparameters:
Classifier

- Batch size: 64

- Optimizer: AdamW optimizer

- Learning rate: $1e-5$

- Learning rate decay parameters: 0.99, 0.998, 0.999, 0.9998 for classifiers trained on 100% of the training set, 5%, 1%, 0.5%, 50 samples, and 20 samples, respectively. The decay parameter was set to 0.99 during epochs with presumptive labels during semi-supervised training.

- Weight decay rate: $1e-7$

- Beta parameters: $(0.9, 0.999)$

Conditional Generative Flow

- Batch size: 320 across 4 GPUs

---

[1]The CT scan dataset can be accessed at `www.kaggle.com/hgunraj/covidxct/version/1`

- Optimizer: AdamW

- Learning rate: $5e - 4$

- Learning rate decay: It had a warm-up period of 10 epochs and was decayed on an exponential schedule with decay parameter 0.99.

- Weight decay: $1e - 6$

- Beta parameters: $(0.5, 0.999)$

- Temperature for Gaussian Noise Sampling: 0.9

Network Architecture
Classifier Our classifier network is based on COVIDNet, by [156]. It is composed of lightweight projection-expansion-projection-extension (PEPX) modules. The PEPX modules consist of 1×1 convolutions for first stage projection that projects input features to a lower dimension, 1×1 to expand the features to a higher dimension different than that of the input features, a depth-wise representation of features to learn spatial characteristics with 3×3 convolutions, 1×1 convolutions to project features back to a lower dimension and finally 1×1 convolutions to extend the channel dimensionality to produce the final features. We take the dimension of the conditional input (z) to be 32, and perform l_2-normalization on it before feeding it to the conditional generative flow. The network architecture also leverages selective long-range connections. One disadvantage of long-range connections is complexity and memory overhead so we use these long-range connections sparingly, this is exhibited by the existence of only four densely connected convolution layers. The network design choices allow COVID-Net to achieve high representational capacity and improves ease of training while maintaining computational and memory efficiency.

Conditional Generative Flow We use a variant of Glow [134] model that features a reorganized flow step, designed to reduce the number of invertible 1×1 convolutions, together with a fine-grained multi-scale architecture. Each coupling layer consists a 3×3 convolution with ELU [157] non-linearity, a 1×1 convolution, a channel-wise summation with a condition vector, a non-linearity, and a final 3×3 convolution. The condition vector is obtained by taking the embedding of the image at the penultimate layer of our classifier and projecting it to the hidden dimension of the 1×1 convolution layer.

We use a 4 level flow, with granularity factor $M = 4$. The first and last levels consist of 8 flow steps, and the two internal levels each consist of a sequence of 3 blocks of 8 flow steps. The hidden dimension of the affine coupling layer at each level is $24, 512, 512, 512$.

Computation: We used 4 NVIDIA Tesla V100 GPUs for the experiments.

Testing Procedure

We performed both quantitative and qualitative testing for conditional synthetic data generation by our model. A test set is held out from the real dataset to be used for quantitative
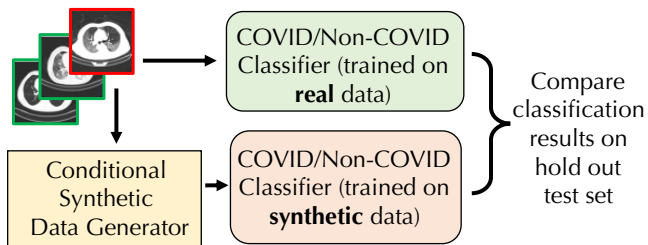
Model	FID
[151]	0.2504
ACGAN	0.0986
CAGlow	0.0483
Ours	**0.0077**

Figure 5.3: Illustration of quantitative testing procedure for conditional synthetic generation.

Table 5.2: Qualitative (Fréchet Information Distance) scores for synthetic data generated by various models (the lower the better).
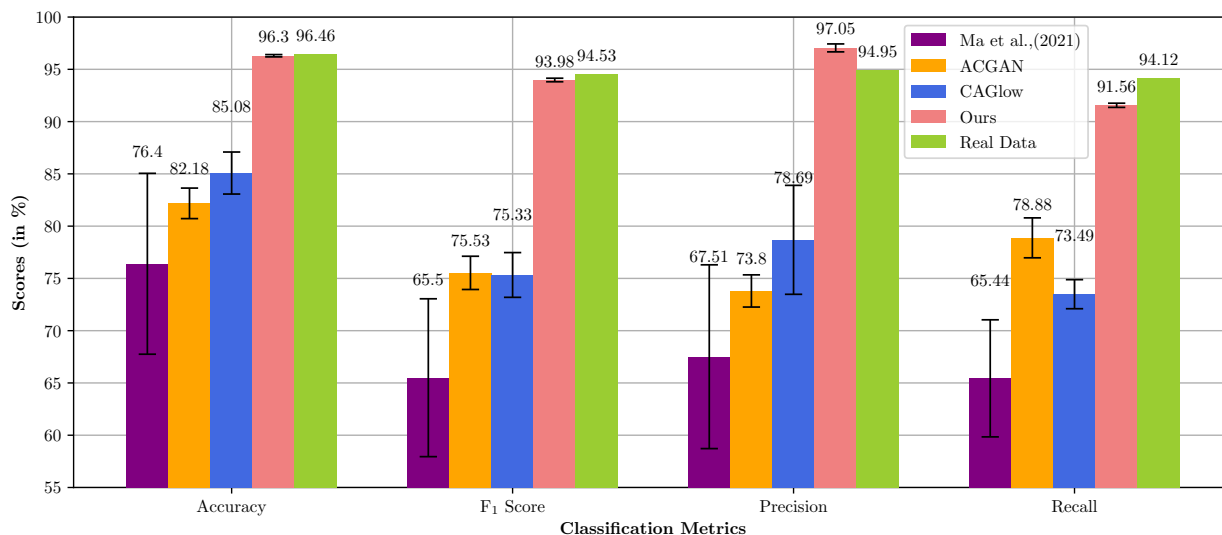
Figure 5.4: Classification metrics for classifiers trained on synthetic data generated by various models. The error bars indicate the variation in classifier performance when the synthetic datasets used to train them were generated multiple times with different seeds. Real data classifier does not involve multiple synthetic data generation, so its error bars are not included.

testing. We then compare the classification performance (COVID/Non-COVID) on this test set for a classifier trained on real data vs a classifier trained on the generated synthetic data. This testing procedure is illustrated in Fig. 5.3. Since the datasets are imbalanced, we report the precision, recall and macro-F_1 score (together referred to as classification metrics) along with the accuracy. For more information on the metrics, please refer to [158]. Closeness of the classification metrics of classifiers trained on synthetic and real data indicates an efficient design of the conditional synthetic generator. To evaluate the quality of generated samples, we report the Fréchet Inception Distance (FID) [159] for the synthetic samples. For FID calculation, we use the embeddings from our classifier trained using real data, in place of the official inception network [160], since the latter is not trained on medical imaging data.
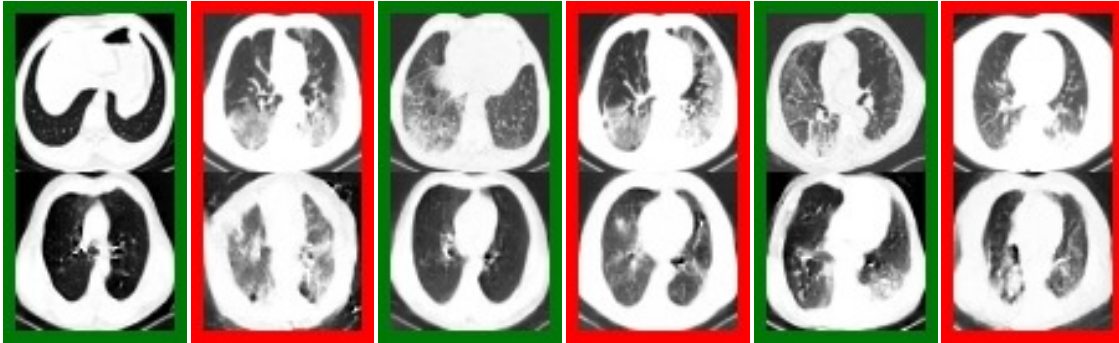
Figure 5.5: Original and generated synthetic CT scan samples. The top row consists of original samples, and corresponding image in the bottom row is the synthetic sample obtained by preserving the original conditional feature representation, and varying the local noise. Image pairs with a red border: COVID samples, and a green border: Non-COVID samples.

Results: Conditional Synthetic Data Generation

The classification results for a classifier trained on the real data vs a classifier trained on purely conditional synthetic data, and tested on a hold-out set of real data, is given in Fig. 5.4. Across the existing methods for conditional synthetic generation, the classifier trained with synthetic data from our proposed model has the closest accuracy, F_1 score, precision and recall to that of the classifier trained on real data. This shows the capability of our method to generate synthetic samples with a distribution that closely matches the real conditional data distribution. The qualitative results (FID scores) for synthetic data generated by various models are tabulated in Table 5.2. The FID scores for our model is the lowest among all models, demonstrating that the quality of the generated samples closely matches the real ones.

It is worth noting that the accuracy/F_1 score of the classifier trained with synthetic data generated by [161] is much smaller as compared to those by other models, not to mention the classifier trained on real data. This can be justified from the fact that [161] relies on an unsupervised method of decoupling global and local information. But for conditional synthetic generation applications, such as the one presented in this paper, the model needs information on what the model designer/ domain experts consider as the conditional information (COVID/Non-COVID in our case). ACGAN and CAGlow have different generators, but both include an auxiliary supervision signal to conditionally guide the generation process. Hence, performance of classifiers trained on synthetic data generated by them are close. We encode the conditions using feature extractors to feed to the generator, leading to state-of-the-art results.

The original samples along with the synthetic samples generated by preserving original conditional feature representation and a different local noise for CT scans are shown in Fig. 5.5. The characteristic features for COVID CT scan samples, i.e., ground-glass opacity

Table 5.3: Results for classifiers trained via semi-supervised learning and tested with different sets of labeled samples and test set bootstrapping. Number of training set samples: 61782.

Amount of labeled data	Accuracy (%)	F_1 Score (%)	Precision (%)	Recall (%)
20 samples	84.84 ± 2.91	76.32 ± 5.24	77.15 ± 4.87	76.35 ± 5.87
50 samples	90.87 ± 1.31	85.86 ± 1.73	86.48 ± 2.68	85.43 ± 1.32
0.5% of training samples	93.90 ± 0.46	90.49 ± 0.61	91.30 ± 1.28	89.8 ± 0.68
1% of training samples	95.06 ± 0.49	92.14 ± 0.69	93.94 ± 1.30	90.62 ± 0.48
5% of training samples	95.80 ± 0.20	93.24 ± 0.28	95.09 ± 0.84	91.23 ± 0.50
100% of training samples	96.30 ± 0.11	93.98 ± 0.17	97.05 ± 0.38	91.56 ± 0.20

are well preserved in the synthetic samples. At the same time, the non-conditional local features, e.g. axial plane position for CT scans are considered as local noise. Since original samples for normal and pneumonia cases are merged together to form a single Non-COVID class, sometimes the corresponding synthetic image for a normal sample is a sample with pneumonia characteristics and vice-versa. This occurs since the conditional model learns to treat them as local information. The ability to decouple the feature representations for given conditions from other information in the data, as exhibited by our model, should be considered the strength of an effective conditional generative model.

Results: Conditional Synthetic Generation under Label Scarcity

Previously, we proposed a semi-supervised learning approach to efficiently generate conditional synthetic samples when the number of samples labeled out of the available pandemic data is less. To test our approach, we retained the assigned label (COVID/Non-COVID) for a few samples, and discarded the label for rest of the samples. The amount of labeled samples was varied from 20 samples to 50 samples to 0.5%, 1%, and 5% of the total training data. The ratio between COVID and Non-COVID samples was maintained among the labeled samples. We conducted the presumptive-labeling and classifier training in an iterative manner, and followed by this, trained the conditional generative flow using the conditional feature embeddings obtained using the feature extractors. We then generated conditional synthetic data using the above trained generative model. To show the robustness of our method, we perform bootstrapping on the test set and repeat our experiments using different sets of labeled samples from the training data. For each model, we also evaluated on multiple synthetic sets generated using random seeds. The results of classification models trained on the synthetic data under different bootstraps and seeds is given in Table 5.3 and 5.4.

As is apparent from the table, using even a few labeled samples, our method is able to achieve results on par with the case when all the labels are available. This further reinforces the strength of our approach in generating conditional synthetic data to rapidly adapt ML

Table 5.4: Results for classifiers trained via semi-supervised learning and tested with multiple synthetic sets generated using random seeds.

Amount of labeled data	Accuracy (%)	F_1 Score (%)	Precision (%)	Recall (%)
20 samples	85.70 ± 0.32	78.65 ± 0.65	77.96 ± 0.49	79.48 ± 1.18
50 samples	90.74 ± 0.77	85.27 ± 0.88	86.93 ± 2.03	83.98 ± 0.68
0.5% of training samples	94.66 ± 0.86	91.41 ± 1.27	93.93 ± 2.02	89.39 ± 0.80
1% of training samples	95.04 ± 0.32	92.00 ± 0.47	94.53 ± 0.88	89.96 ± 0.42
5% of training samples	95.62 ± 0.21	92.95 ± 0.28	95.33 ± 0.77	90.99 ± 0.18
100% of training samples	96.30 ± 0.11	93.98 ± 0.17	97.05 ± 0.38	91.56 ± 0.20

models to a new pandemic at its onset, when there is scarcity of such labels. As expected, at lower levels of labeled data, the uncertainity associated with synthetic data generation is high, as is apparent from Table 5.3, which dies down as we increase the labeled data amount. The uncertainity associated with classification models trained on synthetic set generated by our model using different seeds is low. Both the above observations establish the robustness of proposed method.

An important point to note here is that the closeness of results obtained by utilizing 5% of labels as compared to using 100% of labels do not denounce the importance of the rest 95% of labels. In healthcare, improvement of even 1% of accuracy/F_1 score corresponds to a significant number of samples classified accurately, important especially during a pandemic. Thus, our proposed semi-supervised approach should be considered as a remedy for cases when labels are scarce, not as an alternative to fully-supervised approach.

Example Use of Synthetic Data: Robust Detection of COVID-19 via Data Augmentation

Generated synthetic data can be utilized in a number of downstream tasks. We conduct experiments on one of the tasks: robust detection of COVID-19 via synthetic data augmentation. The training data is inherently highly class-imbalanced, with limited samples of COVID and abundant samples for pneumonia and normal cases. To design a robust COVID-19 detection mechanism under such class imbalance scenario, we augment the training data with synthetic COVID samples generated using the proposed model to increase the % of COVID samples and balance the dataset. The augmentation process and the testing procedure is illustrated in Fig. 5.6. The classification metrics for classifiers trained on the augmented training data are given in Fig 5.7.

Examining the classification results, the classifier trained on augmented training data have better performance as compared to classifiers trained only on limited real training data
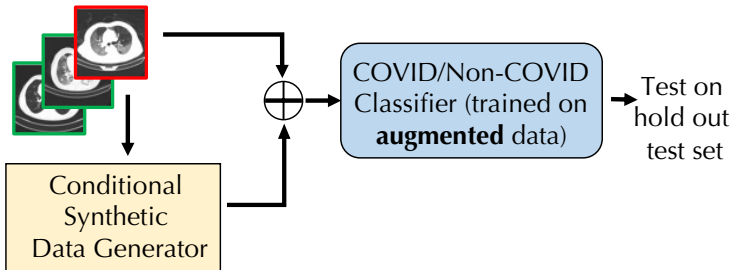
Figure 5.6: Illustration of synthetic data augmentation and testing process. Improvement in performance of classifiers trained on augmented data as compared to that trained on original training data is a step towards robust COVID-19 detection.
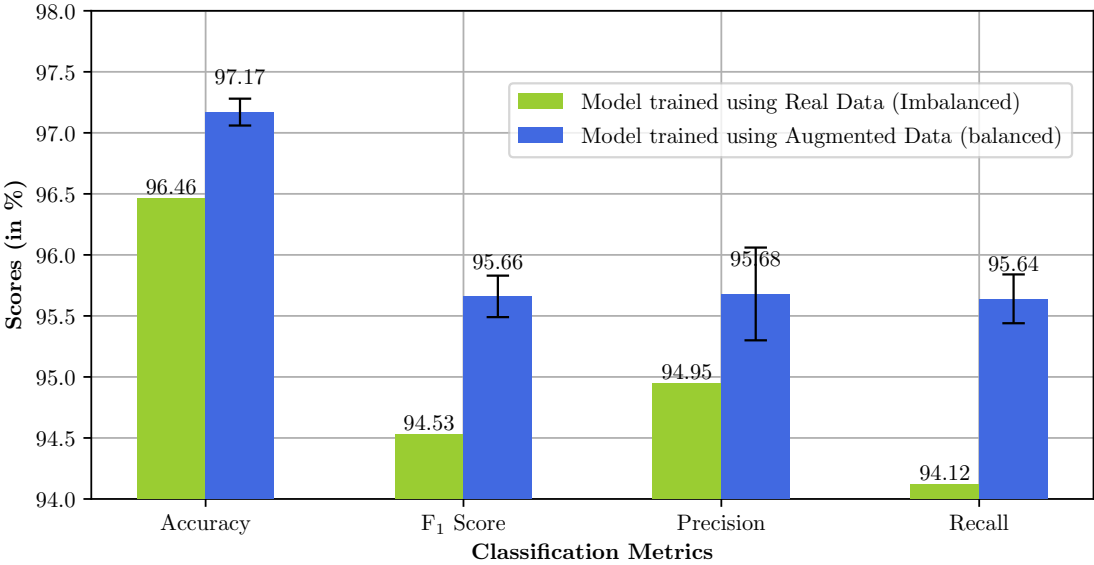
Figure 5.7: Classification results for models trained using real data (with class imbalance) vs augmented data (class-balanced). The real data (having $\sim 20\%$ of COVID samples) was augmented with synthetically generated COVID samples using the proposed model for class balancing.

for all augmentation levels. Note that even slight improvement in the recall score translates to numerous samples classified correctly (e.g. 1% improvement in recall for CT scan corresponds to 200 more correctly classified samples), leading to better diagnosis leading to accurate and timely treatment.

5.4 Related Work

In the field of healthcare, synthetic data generation has been proposed to expand the diversity and amount of the existing training data, often to improve the robustness of machine learning models. [162] propose a generative adversarial network (GAN)-based synthetic data generator to improve the diversity and the amount of skin lesion images. [163] synthesize pathology images for cancer with realistic out-of-focus characteristics to evaluate general pathology images for focus quality issues. [164] propose synthetic generation to produce high-resolution artificial radiographs. In the space of combating COVID-19, [165] propose a method of strengthening the COVID-19 forecasts from compartmental models by using short term predictions from a curve fitting approach as synthetic data. Similarly, [166] and [167] propose a conditional GAN-based generator for synthetic chest X-ray/CT scan data generation and augmentation for robust COVID-19 detection. Above works do not focus on the case where data with proper labels might be unavailable or sparsely available, whereas we tackle this challenge using a semi-supervised approach. We also show the robustness achieved using our model via experiments with several bootstrapping methods.

In the area of conditional generation, a hybrid flow and a GAN-based model have been proposed in CAGlow [168]. In general, GAN-based methods are known to be hard to train [169] and do not provide a latent embedding suitable for feature manipulations [134]. In contrast, we proposed a conditional generation method with efficient decoupling of the conditional information and local noise over an embedding space, along with a flow based generator, which recently have proved efficient in synthetic data generation [170,171]. We compared results for our proposed method over CAGlow and ACGAN for synthetic COVID CT scan generation, and showed improved results.

Decoupling of global and local representation for synthetic generation has been proposed in [151], where the global information is decoupled using a Variational AutoEncoder (VAE) [172]. For conditional synthetic generation, it is necessary that the feature representations salient to the given conditions (COVID/Non-COVID) are decoupled from local noise, which is not guaranteed while extracting the same using a VAE. By employing a classifier network for the same, we ensure the relevant conditional information is not lost into the local noise.

Semi-supervised learning based approaches to enhance classification models has been prominent in domain adaptation tasks, where except for a few samples, knowledge about the labels are generally unavailable in the target domain. A number of domain adaptation models, such as FADA [173], [174], [48], [175] etc. employ few-shot learning approach, leveraging the few labeled data available to make the model efficient. In the space of healthcare, semi-

		Subject ID				
		1	2	3	4	5
Classification Metrics	Cohen's Kappa	28.77%/27%	24.59%/23.12%	19.23%/17.91%	33.65%/31.78%	18.37%/15.49%
	Accuracy	84.3%/79.56%	79.22%/75.76%	63.47%/59.03%	77.19%/77.01%	63.22%/61.42%
	AUC	0.81/0.79	0.8/0.77	0.67/0.62	0.78/0.77	0.76/0.74

Table 5.5: Thermal Preference classification performance with classifiers trained on real and synthetic data. The first number among the pair in each box is performance with a classifier trained on real data, while the second number is with a classifier trained on synthetic data generated by our proposed model.

supervised learning approaches have been used for skin disease identification from limited labeled samples in [176], to enhance X-ray classification in [177] and in COVID-19 detection from scarce chest x-ray image data in [178]. We proposed the use of semi-supervised learning in the space of synthetic data generation, to adapt our proposed generative model to label scarce scenarios, common at the onset of a pandemic.

5.5 Synthetic Data Generation for Personal Thermal Comfort

In our previous work [8], we conducted an experiment to collect physiological signals (e.g., skin temperature at various parts of the body, heart rate) of 14 subjects (6 female and 8 male adults) and environmental parameters (e.g., air temperature, relative humidity) for 2–4 weeks (at least 20 h per day). The subjects also took an online survey, where they reported their thermal sensation (on a scale of -3 to +3) and thermal preference (Warmer, Cooler, No Change) among other parameters.

For this work, we generated synthetic data for the 3 thermal preference classes (Warmer, No Change, Cooler) for 5 of the subjects [63, 179]. We designed fully-connected neural networks for the feature extractor, classifier, and conditional generator blocks. A test set is held out from the real dataset to be used for quantitative testing. We then compare the classification performance on this test set for a classifier trained on real data vs a classifier trained on the generated synthetic data. Since the datasets are imbalanced, we report the cohens kappa, accuracy and AUC score (together referred to as classification metrics).

The classification results for a classifier trained on the real data vs a classifier trained on purely conditional synthetic data, and tested on a hold-out set of real data, is given in Table 5.5. The classifier trained with synthetic data from our proposed model has the close classification performance to that of the classifier trained on real data. This shows the capability of our method to generate synthetic samples with a distribution that closely matches the real conditional data distribution.

5.6   Discussion

We presented a novel conditional synthetic generative model aimed at multiplying the samples of interest at the onset of a pandemic. We conducted extensive experiments on chest CT scan dataset to show the efficacy of the proposed model, and improvements in COVID-19 detection performance achieved via synthetic data augmentation. We also proposed and experimented on a semi-supervised learning approach to efficiently generate conditional synthetic data under label scarce conditions. One of the limitations of our proposed method is that it does not exert selective control over the choice local noise, which can sometimes contain information for important interactions in the data, e.g., in our experiments, we extract conditional information salient to COVID/Non-COVID, whereas the information corresponding to everything else, such as CT scan axial positions, variations of pneumonia etc. are all considered to be the noise for the model. In general, this can be attributed to the way conditional generative models e.g. ACGAN, CAGlow function. By implementing our model for personal thermal comfort synthetic data generation, we demonstrated the generalizability of our proposed model. By doing certain changes, our model can be used to generate synthetic data in a wide range of application domains. There can be numerous variations of synthetic samples that can be created using our model, keeping the conditional information same, hence a potential negative societal impact of our work can be misuse of synthetic data to spread misinformation.

Part III

Tackling Data and Model Inconsistencies

Chapter 6

Improved Tabular Data Pre-Processing Methods

6.1 Introduction

The data obtained in smart buildings can be broadly divided into four classes [180]: occupant data, facility data, enterprise data, and distributed energy resources (DER) data. Occupant data refers to the data collected from occupants pertaining to their occupancy, thermal comfort preferences, energy usage, etc. For instance, to ensure occupants are thermally comfortable in buildings, there is an array of research [8, 21, 181–184] focusing on understanding which parameters affect the thermal preference of an individual or a group, and design physics-based or machine learning based predictors to predict them. The data collected from occupants and their immediate environment include environmental variables [54, 185, 186] such as standard effective temperature, air temperature, relative humidity, and air velocity, occupant specific variables [8, 187] such as clothing level, metabolic rate, and in some cases, physiological signals such as heart rate and temperatures at different key body points. All of the above readings can be taken as instantaneous readings for several subjects, or by performing a field experiment with a set of subjects over a period of time. In both the cases, the data is organized into a tabular form, with the above features as columns and each row representing data at a time stamp for an occupant. Some of the above features are continuous and some discrete. Thermal comfort is a key example of smart building components that prevalently have tabular data [19, 24]. Other occupant data, such as the CO_2 concentration of the return air (used to measure occupancy in buildings [49]), infrared radiation changes using PIR sensors (used to reflect the movement information of objects, and hence detect both occupancy and presence [188]), and energy resource consumption data (used to monitor the usage and encourage energy-efficient behavior by providing incentives [189]), are also organized in the form of tables and hence classified as tabular data.

The second class of data in smart buildings is facility data. This corresponds to the data obtained primarily from and for the various mechanical systems present in the building.

The data collected might be used to optimize the operation of different systems such as the Heating, Ventilation, and Air Conditioning (HVAC), or to diagnose faults in the system for predictive maintenance. For example, for monitoring and opportunistically optimizing HVAC system, the energy consumption, temperature and humidity in different zones [190] in a building are collected. For diagnosing faults in the system, parameters such as flow-rate for water systems, actuator statuses (e.g., valve, pump) [6] etc. are collected. All the above datasets are tabular in nature since they are readings that are coming as a stream with a particular frequency from sensors fitted in various appliances.

The third class is enterprise data, which includes data from software systems governing a smart building. For example, data streams from digital twins of a building might contain synthetic measurements of building parameters [191]. The fourth class is DER data, which comprises of data corresponding to renewable energy (mostly solar) generation and consumption measurements [192], occupant/building energy consumption schedule and patterns throughout the day [193], and data corresponding to demand response programs [194]. All of the above datasets are tabular in nature. In retrospect, we realize that a significant number of datasets collected and utilized by machine learning algorithms in smart buildings are tabular in nature and demand specialized methods for pre-processing.

Data pre-processing is a vital step in the machine learning implementation process since inconsistencies among the diverse features in a dataset can cause any algorithm to be suboptimal. Data pre-processing involves a number of operations, such as data cleaning to get rid of or replace missing and/or noisy data, data transformation to convert the data to a common data type as is warranted by the downstream machine learning model, dimensionality reduction (if needed) etc. There has been significant advances on data cleaning and dimentionality reduction operations in existing research works. However, data transformation, which involves steps such as normalization, encoding and dequantization etc. has not received much attention in the machine learning implementation process especially in applied domains such as smart buildings. Data transformations steps such as normalization are necessary to scale the features to common limits (e.g. min-max normalization), and also to model them to follow a known distribution (e.g. standard/gaussian normalization). At the same time, dequantization of discrete features is also necessary for models to learn the data distribution efficiently. Based on our study, we observed that most of the prior works treat continuous and discrete features alike. The most common continuous feature transformation step in existing works are gaussian or min-max normalization. However, real-life continuous feature distributions comprise of several inherent modes, and many machine learning algorithms are sensitive to the modes present, in which case, above normalization methods prove to be sub-optimal. On the other hand, many prior works do not treat discrete features differently, and just consider them as a special case of continuous features with values present just at the discrete markers. In the best case, a few works convert the discrete features to one-hot vectors, which are again discrete in nature. If we fit a continuous distribution (using ML models such as neural networks since they are smooth function approximators) to these discrete values, the model can learn to achieve high likelihood by placing large spikes at these discrete values, while making the likelihood low everywhere else. This is an unnatural

distribution we would like to discourage our model from overfitting to the discretization.

In this work, we focus on the above challenges for tabular data, and propose the use of two novel data transformation methods (the other steps in data pre-processing that precede data transformation, such as data cleaning are kept the same), namely mode-based normalization for continuous features, and uniform and variational dequantization for discrete features. Dequantization refers to adding noise to the discrete variables before they are fed to the machine learning models. By considering thermal comfort datasets as representative tabular datasets for smart buildings, we show that using our proposed methods for data pre-processing leads to significant improvement in thermal comfort prediction performance as compared to the state-of-the-art model with conventional data pre-processing. Needless to say, the proposed methods, being designed in a generic manner for tabular datasets, extend seamlessly for use by other smart building tabular datasets. To the best of our knowledge, we are the first to propose and conduct an extensive study into the data pre-processing methods for the most commonly found data in smart buildings, i.e. tabular data.

In the following sections, we compare our contribution with the existing works (Sec 6.2), describe the proposed data pre-processing methods (Sec 6.3), conduct experiments to test the efficacy of our approach (Sec 6.4), and conclude with practical considerations and future works (Sec 6.5).

6.2 Related Works

Since we focus on the data transformation step in the whole data processing pipeline, we discuss and compare our proposed methods with data transformation methods used in the previous works. For continuous features, gaussian or min-max normalization have been the gold standard in previous works. For instance, authors in [195] use gaussian normalization or $z-$normalization and apply it to the subjective response data to scale it uniformly and to better determine the overall response trends. In [196], gaussian normalization is used for metadata normalization in design of a dynamic multi-task thermal comfort prediction model. Min-max normalization has also been used in [197] to normalize the data for use in K-nearest neighbor based thermal model. Another work that focuses on study of HVAC control strategies using personal thermal comfort and sensitivity models [198] uses min-max normalization to scale the thermal comfort readings. Authors in [199] use min-max normalization on occupant behavior data to study the influence the same on building energy consumption. There are additional ways for normalization as done in [13], where authors perform normalization of skin temperature (continuous feature) by specifically designing a factor that indicates the unclothed/exposed body surface area. They also show that normalization improves the stratification of thermal classes. In our work, we state the shortcomings of the above methods (Sec 6.3) for continuous features, and propose the use of a novel method, namely, mode-based normalization (Section 6.3). The above method, originally proposed in [200], is used to generate synthetic samples for tabular datasets among other possible applications.

When it comes to transformation for discrete features, not much special attention has
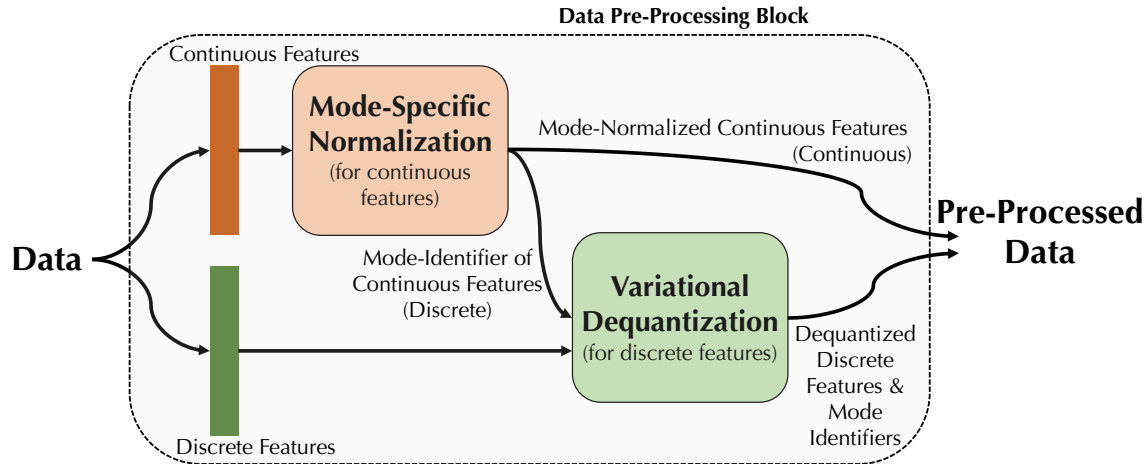
Figure 6.1: Illustration of the proposed data-preprocessing method.

been given to dequantize them before feeding them into machine learning models such as neural networks that are designed to approximate a smooth function with desirable accuracy provided sufficient neurons are used. At best, one-hot encoding has been used to encode categorical variables. For example, Wang et al. [201] study the thermal comfort models designed using ASHRAE database [54], and state that one-hot encoding is commonly used to encode categorical variables such as building type. Authors in [202] also perform one-hot encoding of the categorical features during data pre-processing. Similar is the case for works on data-driven optimization of building designs [203], modeling of energy demand response in buildings [204], etc. This does not only result in high-dimensional data when the categorical variables have many levels, it also gives rise to multiple more variables that are discrete in themselves. To the best of our knowledge based on extensive literature search, there are no existing works that focus on using dequantization methods for discrete feature transformation for machine learning applications in smart buildings. We propose two methods for dequantizing discrete features, namely uniform and variational dequantization [141]. We discuss the ways and cases where the proposed methods can be used, and implement them for a real-life smart building dataset to test for their strength.

6.3 Methodology

In this section, we describe the proposed pre-processing steps for tabular data. Data pre-processing involves a series of steps, such as data cleaning to get rid of or replace missing and/or noisy data, data transformation, dimensionality reduction (if needed) etc. We particularly focus on the data transformation part, keeping the other steps same as others existing in the literature.

Let us assume the dataset in hand is represented by $\mathbf{X} \in \mathbb{R}^{n \times p}$, which means we have n sam-

ples, and p features. The p features are be a mix of both continuous and discrete/categorical columns. Let us represent the continuous feature vectors by $X_1^c, X_2^c, \cdots, X_\alpha^c$, and the discrete feature vectors by $X_1^d, X_2^d, \cdots, X_\beta^d$. Note here that $\alpha + \beta = p$, and each of the above feature vectors have the dimension of $n \times 1$. A continuous feature comprises of values from a continuous domain (e.g., $\mathbb{R}$). A discrete feature takes a value from a discrete set and can either be nominal or ordinal. The number of possible values for each discrete feature can vary among the set of discrete features. Both the continuous and discrete features must be processed in specialized ways for it to be compatible for machine learning (especially neural network) models. Therefore, we propose two data pre-processing methods towards the above goal: mode-specific normalization for continuous features, and variational dequantization for discrete features. An illustration of above pre-processing is shown in Fig. 6.1.

Mode-specific Normalization for Continuous Features

Continuous features in tabular data are usually non-Gaussian and have a number of modes from where the data samples might come from. Gaussian distribution has a single mode, and thus applying transformations that has been used in prior works, such as gaussian or min-max normalization will lead to vanishing gradient problem [200]. Detecting the modes present in the data and using their parameters to normalize the data will help in handling features with complex distributions, a process referred to as mode-specific normalization [200]. In mode-specific normalization, unlike conventional min-max or gaussian normalization, we first detect a mode of the feature distribution from which a particular data sample is highly probable to have come from, and then normalize it with the mean and standard deviation of that particular mode. Post normalization, each feature vector is transformed into two feature vectors, one corresponding to the mode-normalized values which is continuous in nature, and another to the identifier of the mode which was selected for normalization which is discrete in nature. The steps of this process are as follows:

1. A variational gaussian mixture model (VGM) [205] is trained to estimate the number of possible modes for continuous features $X_1^c, X_2^c, \cdots, X_\alpha^c$. For illustration, let us assume for i^{th} continuous feature X_i^c, m number of modes were found. For j^{th} data sample (i.e. j^{th} row of the dataset), the probability of occurrence of the value x_{ij}^c in feature X_i^c is,

$$\mathbb{P}_{X_i^c}(x_{ij}^c) = \sum_{k=1}^{m} \eta_k \mathcal{N}(x_{ij}^c; \mu_k, \phi_k)$$

   where, η_k, μ_k, ϕ_k are the weight, the mean and the standard deviation of mode k.

2. To choose a mode to normalize data x_{ij}^c, we compare the probability of that value coming from each of the possible modes, i.e. mode k^* is chosen for normalization as per,

$$k^* = \arg\max_{k=1}^{m} \eta_k \mathcal{N}(x_{ij}^c; \mu_k, \phi_k)$$

3. Finally, the normalized output and identifier are:

$$\text{Mode-normalized value} \quad = \frac{x_{ij}^c - \mu_{k*}}{4\phi_{k*}}$$
$$\text{Mode Identifier} \quad = k^*$$

We represent the feature vector with mode-normalized values (which is a continuous feature) for X_i^c as X_i^{cc}, and the feature vector with corresponding mode-identifiers (which is a discrete feature) as X_i^{cd}. Effectively, X_i^c is transformed into X_i^{cc} and X_i^{cd}.

Uniform and Variational Dequantization for Discrete Features

Dequantization refers to adding noise to discrete values to make them continuous. Since many of the machine learning models such as neural networks are smooth function approximators, making the discrete features continuous by adding small amounts of noise helps the machine learning model learn the discrete feature distribution efficiently. The distribution from which the noise is extracted brings in the novelty among the dequantization methods. We use two methods for dequantization, namely uniform, and variational dequantization [141]. In uniform dequantization, noise from a compatible uniform distribution is added to the discrete features, whereas, in variational dequantization, the amount of noise that has to be added is dependent on the original data distribution. At this stage, we dequantize the original discrete features that were present in the dataset $(X_1^d, X_2^d, \cdots, X_\beta^d)$, along with the hybrid discrete features that were created as part of the mode-based normalization process before $(X_1^{cd}, X_2^{cd}, \cdots, X_\alpha^{cd})$. Let us denote the union of both the above sets of discrete features as $\tilde{X}^d$.

For dequantization, we add noise $\mathbf{u}$ to the feature set $\tilde{X}^d$, i.e.

$$\tilde{X}_{dequantized}^d = \tilde{X}^d + \mathbf{u}$$

In uniform dequantization, $\mathbf{u}$ is sampled from an uniform distribution $[0, 1]^{\alpha+\beta}$. As it can be observed, the noise added does not have any relation with the data to which it gets added, which although solves the problem of fitting continuous distribution to discrete data but still makes it sub-optimal to learn the data distribution due to the step function in uniform noise distribution. On the other hand, in variational dequantization, $\mathbf{u}$ comes from a variational posterior distribution $q(\mathbf{u} \mid \tilde{X}^d)$. Variational dequantization is powerful as compared to uniform dequantization because the noise added is dependent on the data, hence producing a smooth processed data distribution that is easier for the downstream machine learning model to learn. We model the posterior distribution as a conditional generative flow as $\mathbf{u} = q_{\tilde{x}^d}(\epsilon)$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is gaussian noise. The conditional flow model is jointly trained with the downstream neural network model being trained on the pre-processed data.

We model the conditional flow with coupling transformations as has been proposed in [141]. The coupling transformations (F) are designed to follow the cumulative density function

(CDF) of mixture of M logistic distributions, represented by LMCDF, i.e.

$$F_{LMCDF}(y; \pi, \mu, \mathbf{s}) = \sum_{i=1}^{m} \pi_i \sigma((y - \mu_i) \exp(-s_i))$$

where, $\pi, \mu, \mathbf{s} \in \mathbb{R}^{\dim(y)}$ are the parameters of logistic mixture distribution corresponding to mixture weight, component means, and component scales, respectively, and $\sigma(.)$ denotes the sigmoid function. The input noise vector ϵ is partitioned into two parts, $\epsilon = [\epsilon_1, \epsilon_2]$, as is done for affine flow models. The dequantization noise $\mathbf{u}$ is formulated as,

$$\mathbf{y} = NN_\theta(\tilde{x}^d)$$
$$\pi, \mu, \mathbf{s} = NN_\delta([\epsilon_1, \mathbf{y}])$$
$$\mathbf{u_1} = \epsilon_1, \mathbf{u_2} = F_{LMCDF}((\epsilon_2; \pi, \mu, \mathbf{s})$$
$$\mathbf{u} = \sigma([\mathbf{u_1}, \mathbf{u_2}])$$

where, $NN(\theta)$ and $NN(\delta)$ are neural networks parametrized by θ and δ respectively. We stack multiple such layers in a cascaded manner to generate the dequantization noise $\mathbf{u}$.

An important observation here is that in variational dequantization the networks generating noise are trained in tandem with the downstream model that gets fed with the pre-processed data. Additionally, variational dequantization is designed using neural networks as noise generators. Hence, above method should be used when the downstream model used is a neural network itself that trains using stochastic gradient descent, which essentially holds true for all the deep learning applications in buildings. In cases where the downstream model is not a neural network, uniform dequantization can be a good choice for discrete data transformation.

After the above preprocessing steps, the original data $\mathbf{X}$ becomes,

$$\mathbf{X} = X_1^{cc} \oplus \cdots \oplus X_\alpha^{cc} \oplus X_{1,dequantized}^{cd} \oplus \cdots \oplus X_{\alpha,dequantized}^{cd} \oplus$$
$$\oplus X_{1,dequantized}^{d} \oplus \cdots \oplus X_{\beta,dequantized}^{d}$$

which is then used for downstream tasks such as forecasting, prediction, segmentation or synthetic data generation.

6.4 Experiments

In this section, provide the features metadata of datasets we use, and then share the experimental settings and results.

Datasets

As representative tabular datasets available in smart buildings, we choose two publicly available thermal comfort datasets (obtained from right-here-right-now readings as well as personal thermal comfort field experiments) for testing our pre-processing methods. We test our methods independently for each of the above datasets.

Table 6.1: List of continuous and discrete features for the datasets used in the experiment

Dataset	Continuous Features	Discrete Features
Comfort Database	Standard Effective Temperature, air temperature, relative humidity, air velocity	Clothing level, metabolic rate
Wearables Dataset	Temperature, humidity, wind velocity, physiological parameters: temperature at wrist, ankle, and pant, heart rate	Vote time (morning (7am-12pm), afternoon (12pm-5pm), evening (5pm-10pm), night (10pm-7am)), location during vote (indoors/outdoors)
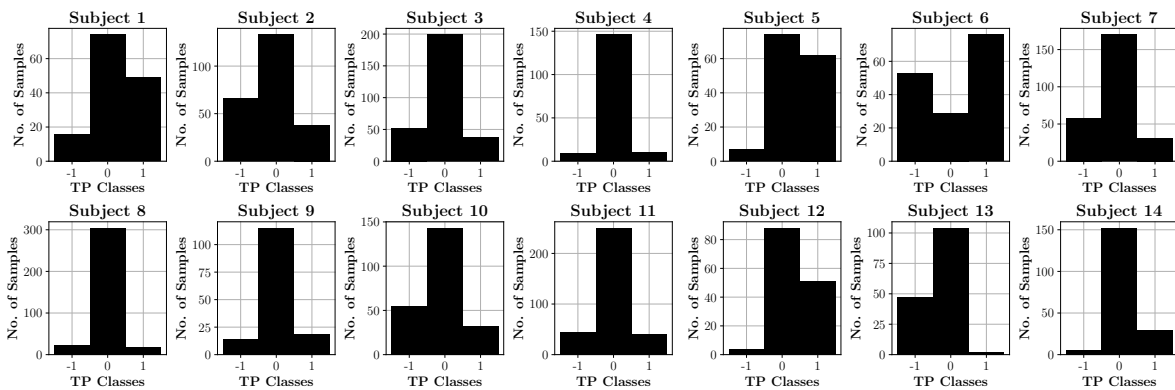
Figure 6.2: Subject wise distribution of data samples in each of the thermal preference classes. Here, "-1"represents "Prefer cooler"class, "0"represents "Prefer no change"class, and "1"represents "Prefer warmer"class.

Comfort Database/ASHRAE Global Thermal Comfort Database II

The ASHRAE Global Thermal Comfort Database II [54], or as we will call "comfort database", is one of the large and mostly used dataset when it comes to designing and testing thermal comfort algorithms, as well as to study the thermal comfort distribution across building types, geographies etc. It is built up of the data from thermal comfort studies conducted around the world in the last two decades from the time the paper was published. It provides thermal comfort measurements, as well as the preference label. We picked six of the most significant variables for data-driven thermal comfort in line with previous researches using this dataset [60]. Specifically, the features chosen are Standard Effective Temperature (SET), clothing level, metabolic rate, air temperature, relative humidity, air velocity. The characteristic type (continuous/discrete of these features is given in Table 6.1. Post data cleaning to get rid of missing values/ NaNs, the total number of data samples remaining was 56148. The distribution of data samples in the three thermal preference classes was "Prefer cooler": 17794, "Prefer no change": 28195, "Prefer cooler": 10159.

Wearables Dataset

We refer wearables dataset to the data collected from personal thermal comfort experiment using wearable sensors by Liu et al. [8]. The authors have performed feature engineering to obtain the mean, standard deviation and gradient of physiological features over last 5 mins, 15 mins, and 60 mins of the vote time, which we use in our work. We ranked the features in the dataset as per the amount of missing values/NaNs existing in them, and got rid of those with large number of missing values. After data cleaning, we had approximately 210 samples available per subject. We also converted the vote time variable to a categorical variable as per the following mapping: "Morning"(7am to 12pm), "Afternoon"(12pm-5pm), "Evening"(5pm-10pm), "Night"(10pm to 7am). The distribution of continuous and discrete features that we use for experimentation using this dataset is given in Table 6.1. The subject wise distribution of data samples in each of the thermal preference classes is shown in Fig 6.2. As it can be observed, the dataset for every subject is highly class-imbalanced with the "Prefer no change"class being the most frequent class.

Experimental Settings

Testing Procedure:

For the comfort database, we designed classifiers to classify the thermal preference classes. For the wearables dataset, we designed personal thermal comfort models (specific to each subject) to classify their individual thermal preference. As per standard practice [8], for each classifier, we conducted 5-fold cross validation repeated 20 times to estimate the average predictive performance. We report the classification accuracy. Since the datasets are highly class-imbalanced, accuracy alone is not a correct representative of classification performance. So, along with accuracy, we report the cross-validated macro F-1 score [158].

Table 6.2: Thermal preference classification performance with standard deviation bounds for comfort database using various machine learning models and data pre-processing methods.

Data Pre-processing Method	ML Models	Accuracy (%)	F-1 Score (%)
Gaussian normalization for continuous features and One-hot encoding for discrete features (Conventional Method)	LDA	53.8 ± 0.4	38.9 ± 0.5
	K-Nearest Neighbors	52.8 ± 0.4	46.7 ± 0.5
	Gaussian Naive-Bayes	52.5 ± 0.4	43.1 ± 0.5
	Extra Trees	57.1 ± 0.5	50.1 ± 0.5
	Random Forest	57.2 ± 0.5	50.1 ± 0.5
	Neural Network	59.7 ± 0.7	53.4 ± 0.8
Mode-based normalization for continuous features and uniform dequantization for discrete features (Our Work)	Neural Network	61.3 ± 0.6	57.5 ± 0.6
Mode-based normalization for continuous features and variational dequantization for discrete features (Our Work)	Neural Network	63.6 ± 0.6	59.9 ± 0.4

Machine Learning Models and Data Pre-Processing:

We experimented with a number of machine learning models ranging from kernel based and tree based methods, to neural networks. Specifically, we use Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Gaussian Naive-Bayes (GNB), Extra Trees, Random Forest, and feed-forward neural networks. Based on our literature review, we found that random forest (also a tree-based classifier) performs at par or better as compared to Gradient Boosted Trees (GBM) or Extra Trees algorithm [8], which is why it is considered the state-of-the-art in thermal preference prediction models. Hence, we used random forest as a representative algorithm for tree-based model family. Neural Network models, owing to the way they are designed and trained (using Stochastic Gradient Descent), are compatible with a range of advanced machine learning algorithms such as transfer learning- adversarial domain adaptation, synthetic data generation, variational inference etc. Implementing neural network models thus opens the door to otherwise impossible enhancements from the machine learning world that can be used to improve algorithms in smart buildings. We use gaussian normalization for continuous features, and one-hot encoding for discrete features as the baseline pre-processing methods, as the above choice is commonly used for tabular data pre-processing in existing works. We then test our proposed pre-processing methods: mode-based normalization for continuous features, and uniform/variational dequantization for discrete features along with the neural network models, and compare them against the above baseline. The neural network architecture for the classifier was kept the same between the baseline pre-processing method, and our proposed methods. For variational dequantization, we use 4 layers of flow models with each layer having feed-forward neural networks representing the NN as mentioned in Sec. 6.3. For wearables dataset, we report results from random forest as the kernel-methods baseline (since it is considered as the state-of-the-art model for thermal preference prediction), and neural network models for a better presentation of the results across multiple subjects. We run the neural network models in a NVIDIA V100 GPU, and use Adam optimizer with a learning rate of $1e - 4$.

Results

The classification metrics accuracy and F-1 scores with their standard deviation bounds for different machine learning models combined with different data pre-processing methods for comfort database are given in Table 6.2. Among the kernel and tree-based methods, it can be observed that random forest performs the best in terms of accuracy and F-1 score among other models. With a feed-forward neural network, which comes with better expressivity potential, while keeping the data preprocessing method the same, we see a 4.37% relative improvement in accuracy, and a 6.59% relative improvement in F-1 score as compared to the random forest results. With our proposed pre-processing methods, mode-based normalization for continuous features, and uniform dequantization for discrete features along with the same neural network model, we see a relative performance improvement of 7.17% in accuracy and a significant 14.77% improvement in F-1 score over random forest. It is to be expected

because effectively by dequantizing and normalizing, we are smoothing the distribution for the continuous neural network models to learn. In the above combination, if we replace uniform dequantization with variational dequantization, we observe a relative improvement of 11.19% in accuracy, and a 19.56% improvement in F-1 score over random forest. This improvement in scores is indicative of the potential of the proposed data transformation methods for tabular data.

In the case of wearables dataset, we designed personal thermal comfort predictors using the above machine learning models. The accuracy and F-1 scores for various models for each subject is given in Fig 6.3. Across all subjects, the average relative improvement over random forest in accuracy was 0.72%, and in F-1 score was 2.79% for a neural network model with gaussian normalization for continuous features, and one-hot encoding for discrete features. When we implemented our proposed mode-based normalization, and uniform dequantization, the average relative improvement over random forest increased to 2.71% in accuracy and 7.33% in F-1 score. Finally, with mode-based normalization and variational dequantization with a neural network model, we observed the highest average relative improvement over random forest: 4.51% in accuracy and 11.22% in F-1 score. It can be observed that the improvement in F-1 score with our proposed methods is significant as compared to that in accuracy. It can be attributed to better encoding of the minority classes, an added benefit for imbalanced datasets commonly found in smart buildings. An important observation to note is that for subjects 4,8,9, and 14, the classification accuracy degrades with the implementation of neural networks and proposed pre-processing methods. One of the reasoning for the the same can be the extreme class-imbalance found in thermal preference classes for those subjects as observed in Fig 6.2. The ratio between sum of all the minority classes and the single majority class for these subjects is as high as 1:7. However, the F-1 score always improves with implementation of proposed pre-processing methods. Since our methods are specifically designed for neural networks and not random forest models, a fair separation and ablation study of machine learning models and the pre-processing method to understand the contribution of each towards the improvement/degradation is difficult in this particular case. However, keeping the neural network model fixed, when we implement gaussian normalization, mode-based normalization + uniform dequantization, and mode-based normalization + variational dequantization, the classification scores increases in that order across all of the subjects. This proves that the combination of the above proposed methods is beneficial for tabular data pre-processing in smart buildings. The choice of transformation method to be used depends on the particular application, and the machine learning models that are planned to be implemented (Sec 6.3).

6.5 Conclusion and Future Work

In this research, we proposed the use of several novel data transformation methods for use in tabular data pre-processing, namely mode-specific normalization (for continuous features), and uniform and variational dequantization (for discrete features). We conducted experimental analysis of thermal comfort prediction models (both group-based and personal thermal

comfort) with the above data pre-processing methods, and showed significant improvement in classification accuracy and F-1 score as compared to state-of-the-art results. In Sections 6.3, and 6.3, we also summarized the scenarios when the above methods can be used. Focusing on the practical usability of our methods, all the pre-processing methods we proposed, except for variational dequantization are compatible with both kernel-based (LDA, KNN, GNB, RF, GBM) and neural network models. However, the variational dequantization is only compatible with neural networks. Hence, the choice of pre-processing method for discrete features should be made based on the machine learning model (kernel-based/neural network) chosen for the downstream task. With the above consideration taken into account, since the methods proposed are generalizable for any tabular data, they can be seamlessly used for any smart building tabular dataset, and can aid in efficient machine learning system design.

In the current work, we mainly focused on one of the main classes of structured data found in smart buildings, namely tabular data, and conducted experiments on some representative datasets. A line of future work include the study of performance improvement by using the proposed pre-processing methods in several other smart building and energy system machine learning tasks, e.g. time-series based energy use forecasting [206], demand response [207], occupancy and activity detection [45, 48, 53], HVAC control [208], building retrofits [122], or synthetic data generation for several building related datasets [209] etc. As stated in this work, the methods can be used with certain modifications for other structured data such as graphical data, and unstructured data such as images. Hence, another line of research can be to study the required modifications, and implementations for use cases in smart buildings and energy systems involving such datasets, e.g. power transmission in grids organized as graphs, and satellite imagery for buildings etc.
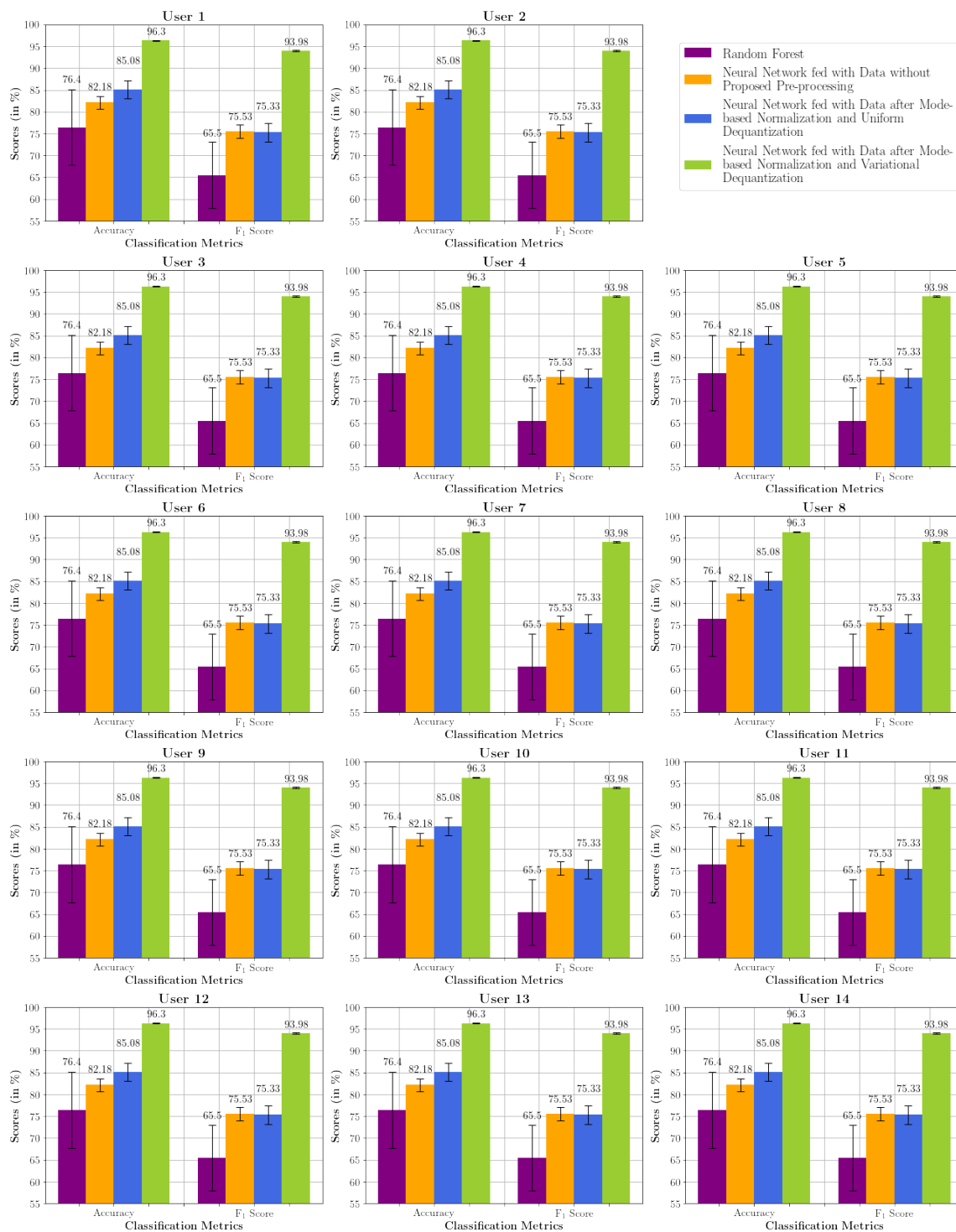
Figure 6.3: Personal thermal preference classification performance with standard deviation bounds for various ML models and data pre-processing methods.

Part IV

Transfer Learning: Cross-Domain Prediction and Generation

Chapter 7

Cross-Domain Classifier Adaptation

7.1 Introduction and Related Works

Humans spend more than 90% of their day indoors, where their well-being, performance and energy consumption are demonstrably linked to thermal comfort. But, study shows that only 40% of commercial building occupants are satisfied with their thermal environment [210]. There has been significant amount of research done to develop models to accurately predict thermal comfort metrics for occupants in a building. Contrary to conventional group-based thermal comfort models, personal thermal comfort models [8] focus on developing thermal comfort predictors at a building occupant level. They have proved effective in human-centric cyber-physical systems to efficiently regulate the building control systems, as well as to understand the correlation between human factors affecting comfort. The general process is to conduct experiments with human subjects and collect their physiological signals along with other environmental parameters, and thermal sensations and preference, to develop models to predict them. In general, such labels are hard to obtain for general occupants in a building. At the same time, the above developed models for experimental subjects do not generalize very well to others, as we will show later in the experiments. We propose an adversarial domain adaptation based method to transfer the knowledge from subjects with thermal preference labels available (hereby referred as the *source*) to those without the labels available (hereby referred as the *target*) and develop a thermal comfort model for the target occupant in an unsupervised manner.

Transfer learning for thermal comfort prediction has been studied at a city-level in [211]. Authors in [212] study transfer learning for personal thermal comfort, but do not focus on underlying assumptions on domain relatedness or few-shot learning cases. Adversarial domain adaptation has been extensively studied in various spaces, such as computer vision and smart buildings [48, 213].
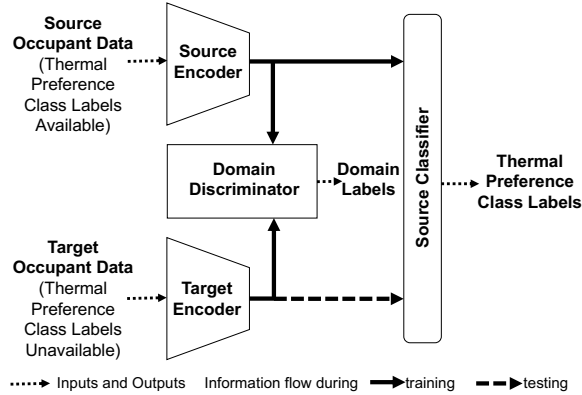
Figure 7.1: Schematic diagram of the proposed method

7.2 Methodology

The objective of this work is to improve the generalization capability of a personal thermal comfort classifier across multiple occupants without collecting labeled data for target occupant(s). Without loss of generalization take the case of two domains, a single source occupant, and a single target occupant. We start by training the source encoder and classifier end-to-end using supervised data from source. For transfer learning, we embed the data from both the domains into a common feature space, and align the target encoder with the source via ADA, in a completely unsupervised manner. At equilibrium, the target and source encoders are aligned, so the previously trained source classifier can be used along with the target encoder for testing in the target domain. The schematic is illustrated in Fig. 7.1, and the training steps are summarized below.

Step 1: Suppose N_s samples X_s with labels Y_s are collected in the source domain with L possible classes. Train a source encoder M_s and a source classifier C_s

$$\min_{M_s,C_s} -\mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} \sum_{l=1}^{L} \left[\mathbb{I}_{[l=y_s]} \log C_s(M_s(X_s)) \right]$$

Step 2: Train a target encoder M_t with X_t unlabelled target samples and fine-tune the source encoder M_s adversarially with a domain discriminator D. Discriminator loss:

$$\min_{M_s,M_t} \max_{D} \mathbb{E}_{x_s\sim X_s} \left[\log D(M_s(x_s)) \right] + \mathbb{E}_{x_t\sim X_t} \left[\log(1 - D(M_t(x_t))) \right]$$

Encoder loss for M_s (similarly for M_t with target data X_t):

$$\min_{M_s} -\mathbb{E}_{x_s\sim X_s} \left[\log D(M_s(x_s)) \right]$$

Source Subject ID

Target Subject ID	1	2	3	4	5	6	7
1	68.35	44.60/53.24	54.68/53.24	53.24/53.24	53.96/53.24	28.78/33.81	44.60/53.24
2	42.62/52.32	66.67	53.59/56.12	56.12/56.12	29.54/56.12	25.74/19.83	51.48/56.12
3	63.54/69.10	60.76/68.06	85.42	69.10/69.10	30.90/69.10	23.61/13.54	64.58/69.10
4	63.64/88.48	86.67/88.48	81.21/88.48	89.09	50.91/88.48	5.45/6.06	81.82/5.45
5	55.24/51.75	48.25/51.75	50.35/51.75	51.75/51.75	72.73	23.78/4.90	53.15/51.75
6	36.08/48.10	30.38/31.65	18.35/18.35	18.35/18.35	32.28/18.35	64.56	18.99/18.35
7	37.07/12.36	62.93/65.64	59.85/65.64	65.64/65.64	42.47/16.60	24.32/11.97	80.69

Figure 7.2: Comparison of thermal preference classification accuracy on target data using a trained source encoder+classifier vs a transfer learning based target encoder+source classifier. Green/Red blocks: Accuracy increases/decreases after ADA.

With aligned the target encoder, thermal comfort prediction for the target domain can be done using the target encoder $+ C_s$.

7.3   Experimental Study

In our previous work [8], we conducted an experiment to collect physiological signals (e.g., skin temperature at various parts of the body, heart rate) of 14 subjects (6 female and 8 male adults) and environmental parameters (e.g., air temperature, relative humidity) for 2–4 weeks (at least 20 h per day). The subjects also took an online survey, where they reported their thermal sensation (on a scale of -3 to +3) and thermal preference (Warmer, Cooler, No Change) among other parameters.

For this work, we developed deep learning based thermal preference classifier for 7 of the subjects, specifically using fully-connected neural networks for the encoder and classifier blocks. We consider permutations of subjects as source and target, i.e. (source,target) $= (i, j), i, j \in \{1, 2, \cdots, 7\}, i \neq j$, to test the extent of transfer learning between the subjects. We start by training a classifier for the source subject, and then align encoders for source and target as described in Sec. 7.2, finally performing testing in target domain using the aligned target encoder and previously trained source classifier. As a baseline, we directly test the source classifier in the target domain, without any knowledge transfer between the domains. The thermal preference classification accuracy is summarized in Fig. 7.2. We observed that for majority of the source/target pairs, the classification accuracy improves after there is transfer learning between the domains, and we are able to design a thermal comfort predictor in the target domain without using any labels. But, for some source/target pairs, the accuracy diminishes.

7.4 Discussion and Future Work

While aligning multiple domains, there is an inherent assumption that the domains share the same set of features at a common feature space. The objective is to obtain a space in which the domains are close to each other while maintaining good performance on the source labeling task. For the above case, the thermal comfort at a personal level depends on a wider range of feature variations. Under such scenario, it is not guaranteed that models developed for a specific subject/occupant of a building can be adapted to be used for any other occupant. This is empirically proved by the diminishing accuracies for some source/target pairs. The underlying closeness between various subjects at a common feature space must be established before adapting thermal comfort classifiers.

This work has a number of future directions, as summarized below.

- Thermal comfort datasets are inherently class-imbalanced. Since adapting personal thermal comfort model of one subject to another does not necessarily lead to improved performance, we conducted a comparison for accuracies. A similar comparison can be done for metrics that reflect imbalance, e.g. F-1 score.

- Domain adaptation can be studied at a group level as source to person level as target and vice-versa.

- Few-shot transfer learning, where few of the target samples are labeled, improving target classification model can be studied.

- Domain adaptation can be studied for cases where only some of the features have labels available, e.g. publicly available features such as room temperature etc.

Chapter 8

Cross-Domain Conditional Synthetic Generation

8.1 Introduction

Prior works on cross-domain translation involve construction of a mapping between two (or more) unpaired domains. The translation consistency is maintained by introducing some form of inductive bias terms such as cycle consistency [214], semantic consistency [215], entropic regulation [216] etc. Most of the proposed models for domain translation are generative adversarial network (GAN) [217] based and involve many-to-one/one-to-many mappings, making the cycle consistency only approximate. A recent work, Alignflow [218] achieves exact cycle consistency by modeling the domains with normalizing flows via a common latent space. Normalizing flows [1, 134] are a class of generative models which map an unknown and complex data distribution to a latent space with a simple (e.g. standard gaussian) prior distribution via invertible mappings. Another benefit with having flow model mappings is that they offer a rich latent space, which is suitable for a number of downstream tasks, such as semi-supervised learning [125], synthetic data augmentation and adversarial training [126], text analysis and model based control etc.

Conditional synthesis has been explored by CGAN [219] by augmenting the conditions with the data and processing it via GAN and by ACGAN [125] by introducing an auxilliary classifier for the conditions. This becomes challenging for flow models which are bijective in nature, and hence indirect methods must be adopted to jointly model data and the conditions. [168] propose an encoder-discriminator-classifier-decoder based approach on flow latent space which can generate synthetic samples for a domain by passing its conditions via encoders and the data via a flow network. They show improvements in varying the quality of generated images for handles relating to various features of the dataset.

We present CDCGen, a generative framework that is capable of transferring knowledge across multiple domains and using it to generate synthetic samples for domains lacking information about labels/attributes. We model the label/attribute scarce domain as the

Table 8.1: Comparison of CDCGen with state-of-the-art cross domain translation and conditional syntheis models. Across the board, CDCGen features all the advantages over other models.

Model	Cross-Domain Translation	Cycle Consistency	Independent Conditional Synthesis	Availability of Latent Space Embeddings
XGAN [215]	✓	Approximate	✗	✗
CycleGan [214]	✓	Approximate	✗	✗
[220]	✓	Approximate	✗	✗
Alignflow [218]	✓	Exact	✗	✓
CGAN [219]	✗	–	✓	✗
ACGAN [221]	✗	–	✓	✗
CAGlow [168]	✗	–	✓	✓
CDCGen (ours)	✓	**Exact**	✓	✓

target, and a related domain with available information about its labels/attributes as the source. We model the source and target domain via normalizing flows with a common latent space. For conditional synthesis, we introduce a variant of ACGAN by using it on the learned latent space rather than the data space, and train it with only the data and available labels from the source domain. The features can be manipulated easily in the latent space, which is learnt by the conditional synthesis network. During the inference phase, CDCGen offers independently specifying conditions, encoding them to a common latent space and moving through the inverse flow to generate conditional synthetic samples in the target domain. Table 8.1 summarizes the comparison between CDCGen and other related models for different feature availability. CDCGen comes out to be an amalgamation of all features available among the model selections.

We establish the CDCGen framework and conduct empirical evaluations with benchmarked image datasets. CDCGen shows encouraging performance in domain alignment, as well as conditional generation for all source and target combinations.

8.2   Related Work

We discuss the related work from two perspectives relevant to the CDCGen framework, namely cross-domain translation and conditional synthesis.

Cross-Domain Translation

Cross-domain translation involves construction of mappings between two or more domains, by training on unpaired data samples in both the domains. Such a problem is under-constrained and involves aligning the domains in feature space via mappings. Several research works in this space [214, 215, 222] introduce a form of cycle consistency loss which ensures that by

translating an image from one domain to another domain via mappings and then applying reverse mappings to translate back yields the same image. XGAN [215] uses additional loss terms to incorporate semantic consistency across domains, to match the subspace for embedding from multiple domains and prior knowledge via pre-trained models. However, since all the above models involve GAN based architectures, they lack a latent space embedding useful for downstream manipulation tasks [134]. Moreover, since the mappings are not guaranteed to be invertible, the cycle consistency is only approximate.

Alignflow [218] involves modeling each of the domains via normalizing flow mappings to a common latent space. It has a hybrid training objective constituting both maximum likelihood estimation and adversarial training. Moreover, since flows are invertible mappings, Alignflow achieves exact cycle consistency. However, flow models, by virtue of the training procedure, face a challenge to align domains which are apart in terms of semantics and/or style, apparent from the generated samples quality in comparison with GANs. For CDCGen, we use the best of both worlds: flow model mappings for the domains to a common latent space, along with loss terms useful to align the domains in the embedding space. CDCGen offers a rich latent space, which is further utilized for conditional synthesis in label/attribute scarce domains (Sec. 8.2).

Conditional Synthesis

Conditional generative models have been introduced to generate desired synthetic data by incorporating conditions information in model design. From CGAN [219] which is a modification of conventional GANs and works by feeding the label/attribute information to the generating block, conditional synthesis has seen different algorithmic variations [125, 223, 224]. A notable work, ACGAN [125] employs an auxilliary classifier for the discriminator to classify the class labels. A recent work, CAGlow [168] proposes a variant of ACGAN with an encoder-decoder network, adding ability to model unsupervised conditions. Additionally, above works deal with conditional generation in a single domain. We use a variant of ACGAN over a shared latent space for multiple domains, thereby transferring knowledge from label-rich domains to perform conditional synthesis in label-scarce domains.

8.3 The CDCGen Framework

In this section, we will present the CDCGen framework capable of generating conditional synthetic samples for a domain in an unsupervised setting. We select a domain with availability of information about the labels/attributes (namely source domain) and has shared attributes with the domain for which we don't have information about labels/attributes (namely target domain). Under this setting, the framework consists of two major networks: one for domain alignment and one for conditional synthesis. We consider the case of two domains, but under the assumption of having shared attributes between the source and target domains, proposed method generalizes to multiple domains seamlessly.

Domain Alignment

The first step in CDCGen is to align the source and the target domains. Let the source and target domain be denoted by $\mathcal{D}_s$ and $\mathcal{D}_t$ with unknown marginal densities p_s and p_t respectively. Both the domains are mapped via invertible transformations (normalizing flows) $\mathcal{F}_s$ and $\mathcal{F}_t$ to a common latent space Z, which serves as a shared feature space for alignment. We assume the shared latent space follows a normal gaussian distribution $p(z)$, common for training of most of the state-of-the-art flow models. The relationship between the sample space and latent space can be represented as,

$$\mathcal{D}_s \xrightarrow{\mathcal{F}_s} Z \xleftarrow{\mathcal{F}_t} \mathcal{D}_t$$

Note that the invertible nature of the flow model is helpful in two different ways,

- It provides a mechanism to translate between source and target domains, with invertible mappings $\mathcal{F}_{s\to t} = \mathcal{F}_t^{-1} \circ \mathcal{F}_s$ and $\mathcal{F}_{t\to s} = \mathcal{F}_s^{-1} \circ \mathcal{F}_t$.

- It helps achieve exact cycle consistency (as introduced in CycleGAN [214] to ensure accurate representation of the mappings) between the domains, since $\mathcal{F}_{s\to t} \circ \mathcal{F}_{t\to s} = \mathcal{F}_t^{-1} \circ \mathcal{F}_s \circ \mathcal{F}_s^{-1} \circ \mathcal{F}_t = I$, where I is the identity matrix.

We use a hybrid training objective involving both maximum likelihood estimation and adversarial training. Flow models are trained with an unsupervised maximum likelihood loss, with a normal gaussian prior on the latent space Z. Since there are two flow models involved for the two domains, the maximum likelihood loss is expressed as,

$$\mathcal{L}_{MLE}(\mathcal{F}_s) + \mathcal{L}_{MLE}(\mathcal{F}_t)$$

For cross-domain mappings, adversarial loss terms are introduced. These terms introduce inductive bias required for cross domain translation [214]. We employ critics $\mathcal{C}_s$ and $\mathcal{C}_t$ for source and target domains respectively, which distinguish between real samples (sampled from the same domain) vs. generated samples (obtained via cross-domain mappings). For example, the adversarial loss for source domain can be expressed as,

$$\mathcal{L}_{ADV}(\mathcal{C}_s, \mathcal{F}_{t\to s}) = \mathbb{E}_{x_s \sim p_s}[\log \mathcal{C}_s(x_s)] + \mathbb{E}_{x_t \sim p_t}[\log(1 - \mathcal{C}_s(\mathcal{F}_{t\to s}(x_t)))]$$

We also use a domain-adversarial loss [225] which forces the embeddings learnt by the flow models $\mathcal{F}_s$ and $\mathcal{F}_t$ to lie in the same subspace. This is achieved by training a classifier $\mathcal{C}_{DAL}$ which takes the latent space embeddings for each domain and classifies the sample to be coming from $\mathcal{D}_s$ or $\mathcal{D}_t$. It is trained in an adversarial manner, with a classification loss function $\ell(\cdot, \cdot)$, such as cross-entropy. $\mathcal{L}_{DAL}$ can be expressed as,

$$\mathcal{L}_{DAL}(\mathcal{F}_s, \mathcal{C}_{DAL}) = \mathbb{E}_{x_s \sim p_s} \ell(\mathcal{D}_s, \mathcal{C}_{DAL}(\mathcal{F}_s(x_s)))$$

(a) CDCGen Training Schematic

(b) CDCGen Inference Schematic
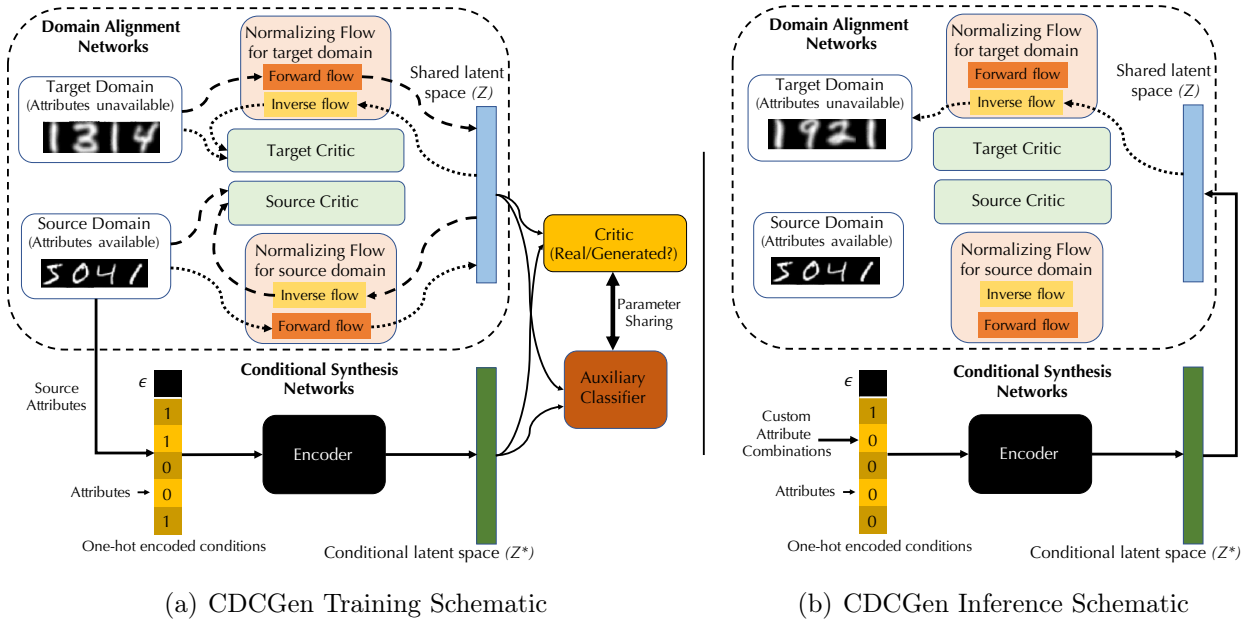
Figure 8.1: Illustration of training and inference methods in CDCGen. The networks inside the dashed box are for domain alignment and those outside are for conditional synthesis.

Finally, for domain alignment, the overall loss term is,

$$\mathcal{L}_{Domain\ Alignment}(\mathcal{F}_s, \mathcal{F}_t, \mathcal{C}_s, \mathcal{C}_t, \mathcal{C}_{DAL}; \lambda_s, \lambda_t, \gamma_s, \gamma_t) = \mathcal{L}_{ADV}(\mathcal{C}_s, \mathcal{F}_{t \to s}) + \mathcal{L}_{ADV}(\mathcal{C}_t, \mathcal{F}_{s \to t})$$
$$+ \gamma_s \mathcal{L}_{DAL}(\mathcal{F}_s, \mathcal{C}_{DAL}) + \gamma_t \mathcal{L}_{DAL}(\mathcal{F}_t, \mathcal{C}_{DAL}) - \lambda_s \mathcal{L}_{MLE}(\mathcal{F}_s) - \lambda_t \mathcal{L}_{MLE}(\mathcal{F}_t)$$

where, hyperparameters λ_s and λ_t dictate the relative contribution of maximum likelihood loss, and γ_s and γ_t correspond to contribution of domain adversarial loss, both as compared to the adversarial loss. The objective is minimized w.r.t. the parameters of the flow models $\mathcal{F}_s$ and $\mathcal{F}_t$ and maximized w.r.t. parameters of $\mathcal{C}_s$, $\mathcal{C}_t$ and $\mathcal{C}_{DAL}$. This procedure is illustrated in the dashed box in Fig. 8.1(a).

Conditional Synthesis

For conditional synthesis, we propose a variant of ACGAN [221]. Instead of using class/attribute conditioning on the sample space as done in ACGAN, we use it in the shared latent space. Under the setting of our problem, we don't have any information about the labels/attributes in the target domain. So, for the conditional synthesis part, only the attributes available from the source domain are used for training.

We denote the available source attributes/conditions as $c_s \sim p(c_s)$, represented as one-hot encodings. Our network consists of an encoder to model the conditions, a critic to differentiate between the real and generated latent vectors, and an auxiliary classifier to

classify the encoded conditions. We will introduce each of the above components and their associated loss functions separately.

Encoder: An encoder network E encodes the conditions (c_s, ϵ) into a latent space Z^* (separate from the shared latent space Z for aligned domains), where ϵ is sampled from standard gaussian distribution $(p(\epsilon))$ and is helpful for incorporating stochastic behavior among condition vectors. Let the distribution for above mentioned latent space be denoted as $p^*(z)$. Our objective is to minimize the Jensen-Shannon (JS) divergence between the encoded distribution $p^*(z)$ and the shared latent distribution $p(z)$ for aligned domains $\mathcal{D}_s$ and $\mathcal{D}_t$. So, the encoder loss is represented as,

$$\mathcal{L}_E = \mathbb{E}_{\epsilon \sim p(\epsilon), c_s \sim p(c_s)}[\log \mathcal{C}(E(c_s, \epsilon))]$$

where, $\mathcal{C}$ is a critic, more about which we describe now.

Critic: A critic $\mathcal{C}$ discriminates between the latent vectors coming from generated conditional distribution $p^*(z)$ and real shared latent distribution $p(z)$ for aligned domains. This is an adversarial loss which is trained so as it is unable to distinguish the latent vectors at equilibrium, thus enabling the encoder E to generate latent vectors close to the real shared latent distribution $p(z)$. The loss function for $\mathcal{C}$ is,

$$\mathcal{L}_{CRITIC} = \mathbb{E}_{z \sim p^*(z)}[\log \mathcal{C}(z)] + \mathbb{E}_{z \sim p(z)}[1 - \log \mathcal{C}(z)]$$

Classifier: A classifier takes the latent vectors $(z \sim p^*(z)$ and $z \sim p(z))$ as input and classifies the conditions (c_s). The classifier loss is a cross entropy loss between the predicted and true conditions. If the class posterior probabilities are $q(c_s|z)$, the classifier loss function can be expressed as,

$$\mathcal{L}_{CLASSIFIER} = \mathbb{E}_{z \sim p^*(z), c_s \sim p(c_s)} q(c_s|z) + \mathbb{E}_{z \sim p(z), c_s \sim p(c_s)} q(c_s|z)$$

The overall loss function for the conditional synthesis part is,

$$\mathcal{L}_{Conditional\ Synthesis} = \beta_E \mathcal{L}_E + \beta_{Cr} \mathcal{L}_{CRITIC} + \beta_{Cl} \mathcal{L}_{CLASSIFIER}$$

where $\beta_E, \beta_{Cr}, \beta_{Cl}$ are hyperparameters. The critic and the classifier networks share their parameters except for their output blocks. Conditional synthesis procedure is illustrated in Fig. 8.1(a).

Inference

CDCGen can generate conditional samples in the target domain, even when the training process does not utilize its class/attribute information. To generate samples with conditions $\tilde{c}_s$, a latent vector $\tilde{z}$ is generated by encoding the one-hot conditions $\tilde{c}_s$ and $\tilde{\epsilon} \sim p(\epsilon)$ via the encoder network, i.e. $\tilde{z} = E(\tilde{c}_s, \tilde{\epsilon})$. Then the latent vector $\tilde{z}$ is passed via the inverse flow $\mathcal{F}_t^{-1}$ to generate the desired sample in the target domain, i.e. $\mathcal{F}_t^{-1}(\tilde{z})$. The inference schematic is illustrated in Fig. 8.1(b).

8.4   Experiments

In this section, we empirically evaluate CDCGen for synthetic generation in label scarce domains.

   Datasets: We perform experiments on 2 standard image datasets for digits, namely MNIST [226] and USPS. MNIST contains $60,000$ training and $10,000$ test images with ten classes corresponding to digits from 0 to 9. USPS has 7291 training and 2007 test data with the same classes as MNIST. To address this imbalance, for each domain, we sample 542 images from the original training set for each class to form the new training set. To form the test set, we sample 147 images from the original test sets for each class. We resize all the images to 32×32 for training and synthesis.

   Source and Target Domain Combinations: We consider two cases, first with MNIST as the source and USPS as the target domain, and second, with the roles interchanged, i.e. USPS as the source and MNIST as the target. We report results for domain alignment and as well as subsequent conditional synthesis in the target domain, all while not using any labels from that domain.

   Networks: We use architecture from RealNVP [1] for each of the domain flows ($\mathcal{F}_s$ and $\mathcal{F}_t$). Typical configurations for RealNVP can be specified as a tuple comprising N_{scales} (number of scales), $N_{channels}$ (number of channels) in the intermediate layers, and N_{blocks} (number of residual blocks in the scaling and translation networks of the coupling layers). For MNIST $\leftrightarrow$ USPS case, both $\mathcal{F}_s$ and $\mathcal{F}_t$ are set to RealNVP(2, 64, 8). The critics ($\mathcal{C}_s$ and $\mathcal{C}_t$) used convolutional discriminators from PatchGAN [227] , each with 16 filters in the critic's first convolutional layer. For conditional synthesis, we concatenate the one-hot vector of labels with components of random noise as input to the encoder. The vector then passes through one fully-connected layer and eight transposed convolutional layers with upsampling scale 2, 2, 2, 2, 2, 1, 1, 1 and channel sizes 256, 1024, 512, 256, 128, 64, 32, 16 respectively. The supervision block contains four convolutional layers with stride 2 and channel sizes 64, 128, 256, 512. This is followed by two separate fully-connected layers for each network head, one for outputting probabilities of real or fake and the other for classifying the label.

   Optimizer: For training the domain alignment network, we use the Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$, and learning rate $1 \cdot 10^{-6}$. For training the conditional synthesis network, we use the Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$, and learning rate $2 \cdot 10^{-5}$.

Domain Alignment

In this section we present the results for domain alignment between source and target combinations. Fig. 8.2(a) shows the source MNIST samples and corresponding USPS samples by translating it via the forward source and inverse target flows. The middle sample is visualization of corresponding latent space sample. Fig. 8.2(b) depicts the same with USPS as the source and MNIST as the target. It can be observed that the class identity is preserved with the translation with the style adapted for the target domains. The sharpness of the translated samples are compromised, which is a result of the flow model assigning some
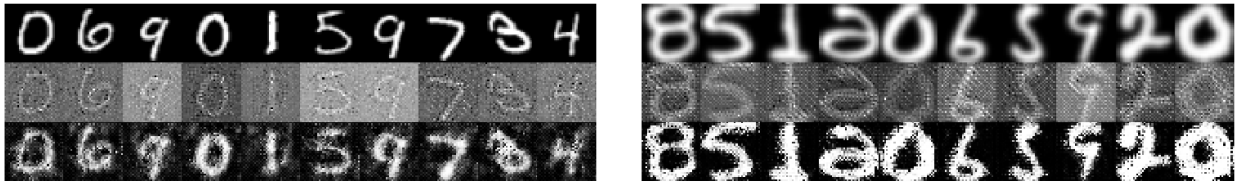
(a) Result with MNIST as source and USPS as target (b) Result with USPS as source and MNIST as target

Figure 8.2: Results for domain alignment between source and target domains. The top row has original samples from the source domain. The middle row is the corresponding latent space mapping and the bottom row is the sample obtained by translating it to the target domain. The USPS images are slightly blurred due to the upscaling applied as standard pre-processing.
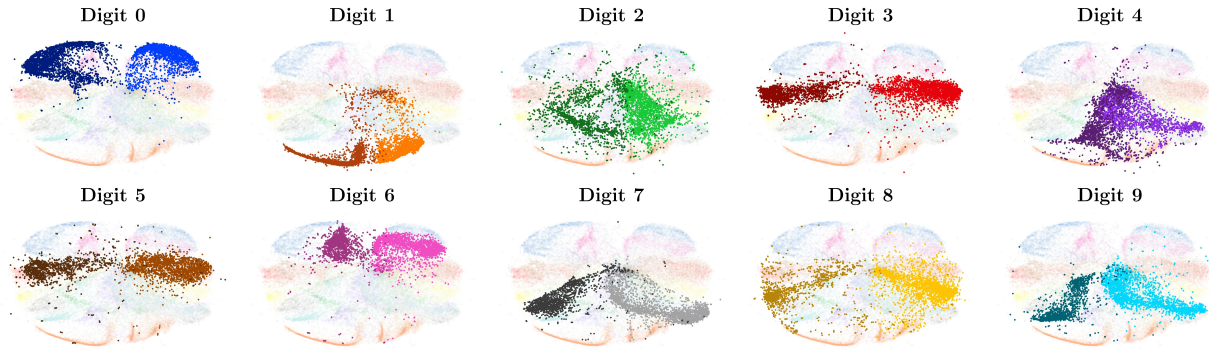
Figure 8.3: t-SNE representation of shared latent space for MNIST ↔ USPS. For each digit, points for USPS are visualized with the darker colors, and points with lighter colors correspond to MNIST.

probability mass to all the samples it is fed. This is unlike pure GAN based models which selectively assign probability mass to meaningful samples.

Another interesting observation is the appearance of digit class identity in latent space visualizations. This is particularly useful from the perspective of CDCGen, since the conditional synthesis network works based on the latent space mappings from both the domains.

We present the t-SNE embeddings for the shared latent space in our proposed domain alignment network for MNIST and USPS in Fig. 8.3. It can be observed that the visualization has distinct clusters for each digit class, but the embeddings from both the source and target domain are close and belong to the same cluster for the overall digit class clustering. The visualization allows us to infer that the latent space has learned a subspace corresponding to each digit, and interpolating across this subspace is effectively a conditional feature-preserving domain transfer.

(a) Generated samples for USPS as target   (b) Generated samples for MNIST as target
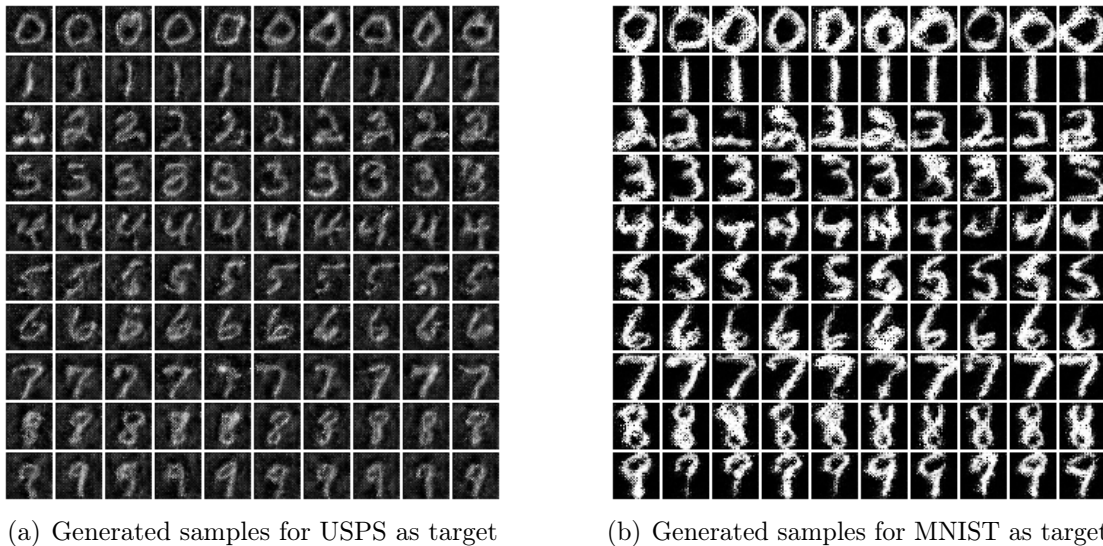
Figure 8.4: Conditional synthetic samples generated by CDCGen. The rows represent conditioned digit classes (0-9) and the columns include more samples for each class.

Conditional Synthesis

We trained the conditional synsthesis part of CDCGen (Section 8.3) with source labels to generate conditional synthetic samples in the target domain. Fig. 8.4(a) shows the samples generated with USPS as the target domain and Fig. 8.4(b) shows the samples generated with MNIST as the target. Each row corresponds to the digit classes which are assigned as conditions. It can be observed that CDCGen is able to generate synthetic samples belonging to the digit class as conditioned. There are also variations among the samples across different columns which shows the stochastic nature of generation by CDCGen. The compromise in sharpness of the samples generated can be observed in the generated samples too, and is owed from the domain alignment mappings by flow models.

Imbalance between Adversarial and Maximum Likelihood loss in Domain Alignment

In this section we present the impact of imbalance between adversarial and maximum likelihood loss terms, in order to highlight the design preferences for CDCGen. The results presented in previous section are such that the maximum likelihood loss is comparable to adversarial loss. For this section, we reduced the λ_s and λ_t terms which correspond to proportion of maximum likelihood loss as compared to adversarial loss for source and target respectively. We also used a Wasserstein loss with gradient penalty [228] for the adversarial losses in domain alignment. This was done so as to selectively give more power to
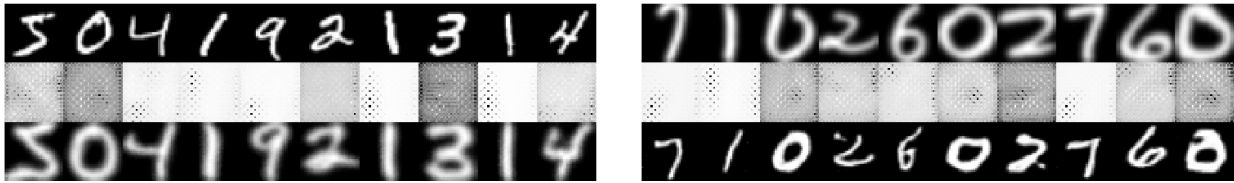
(a) Result with MNIST as source and USPS as target   (b) Result with USPS as source and MNIST as target

Figure 8.5: Results for domain alignment between source and target with less weight on maximum likelihood loss. The top row has samples from the source domain. The middle row is the corresponding latent space mapping and the bottom row is the sample obtained by translating it to the target.

the adversarial loss as compared to the maximum likelihood loss. The samples for domain alignment are presented in Fig. 8.5(a) and Fig. 8.5(b) for MNIST $\rightarrow$ USPS and USPS $\rightarrow$ MNIST respectively. It can be observed that the sample quality has improved substantially, coherent with the fact that GANs are capable of generating qualitative samples. But at the same time, the latent space representations are lacking efficient representation, since the flow model learning objective is underrepresented. For this case, learning the conditional distribution becomes challenging for the conditional synthesis network. Summarizing, the domain alignment network should be designed to have a right balance between maximum likelihood loss (to make the latent space representative) and adversarial loss (for alignment).

8.5   Conclusions

CDCGen, a generative framework capable of generating conditional synthetic samples for domains without the requirement of obtaining its labels/attributes was presented. We also conducted empirical studies with standard image datasets to observe feature transfer and independent conditional generation. In the future, making the conditional generation models across multiple domains can be studied with varying levels of label availability (few-shot learning) for target domain. CDCGen can also be adapted for other modalities of data including audio and tabular data. These can be used in smart buildings for several applications, including synthetic thermal comfort data generation across domain variations such as climate to climate, geography to geography, and one occupant to another, synthetic energy consumption data generation across multiple buildings in a city etc.

Chapter 9

Conclusion and Future Works

9.1 Conclusion

In this research, we explored multitude of ways in which ML has been utilized in smart buildings and pandemic-specific healthcare. Then we identified 3 major data-specific challenges existing in these applications.

Our research focused on tackling the above challenges using generative modeling and un/semi-supervised learning. We proposed conditional synthetic data generation to generate synthetic data and use them in tandem with real data to make ML models robust and efficient. We proposed transfer learning algorithms, namely domain adaptation, and style transfer algorithms to tackle the challenge of domain discrepancy and unavailability of data/labels across multiple domains. Finally, we presented data pre-processing methods to bridge the gap between tabular data commonly found in smart buildings and their use in many state-of-the-art ML models such as neural networks, which are smooth function approximators.

9.2 Future Works

The proposed research opens gate to an wide array of future research. Fig. 9.1 illustrates the methodological advances that can serve as future works for our current work, with potential to be applied smart buildings and other infrastructures.

Generative AI for Multimodal Data/Sensor Fusion

Our research focused on Generative modeling and other ML algorithms for images, tabular, and time-series data in buildings independently. A potential future research area can be applying generative modeling, synthetic data generation, domain adaptation with sensor fusion techniques for multimodal data in smart buildings, especially a mixture of

- 3D mapping data- e.g., from LiDAR sensors
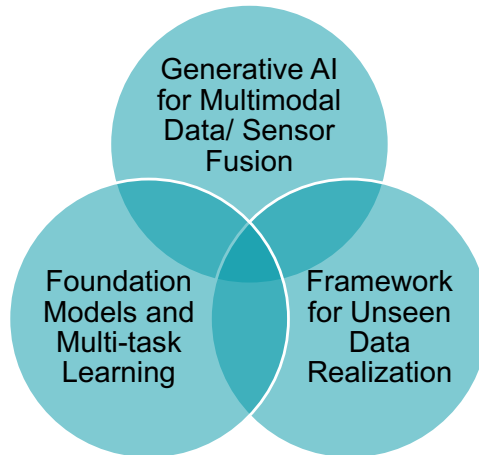
Figure 9.1: Future Avenues of Current Research

- Tabular/time series data from sensors (CO2, Lighting, Plug load)

- Images (RGB/Infrared Cameras)

In one of our past works, we explored this direction by combining data from WiFi systems, and camera. We designed WiVi [53], a novel human activity recognition scheme that is able to identify common human activities in an accurate and device-free manner via multimodal machine learning using only commercial WiFi-enabled IoT devices and camera. For sensing using WiFi, a new platform is developed to extract fine-grained WiFi channel information and transform them into WiFi frames. A tailored convolutional neural network model was designed to extract high-level representative features among the WiFi frames in order to provide human activity estimation. We utilized a variant of C3D model for activity sensing using vision. Following this, WiVi performed multimodal fusion at the decision level to combine the strength of WiFi and vision by constructing an ensembled DNN model. Extensive experiments were conducted in an indoor environment, demonstrating that WiVi achieves 97.5% activity recognition accuracy and is robust under unfavorable situations, as each modality provides the complementary sensing when the other faces its limiting conditions. This research can be extended to other modalities as described above.

Foundation Models and Multitask Learning

Foundation models, also known as pretrained models or base models, are large-scale language models that are pre-trained on massive amounts of data. The data can come from open-source or enterprise data. These models capture semantic and syntactic patterns in the input data. Foundation models are useful for multitasking because they can be fine-tuned or adapted to perform specific tasks by providing additional task-specific training data. Instead of training a model from scratch for each task, which would require a large amount of task-specific

data and computational resources, fine-tuning a foundation model allows for faster and more efficient training.

Foundation models have enormous potential in several applications in smart buildings. Foundation models can be used to capture patterns in an unsupervised way from large corpus of buildings data, and then can be fine-tuned for a number of applications, including,

- Energy Management: analyze energy consumption patterns and optimize building energy management systems, including automated identification of anomalies, and suggestion of strategies for improving energy efficiency.

- Building Design: Automated design of buildings with varying building materials and blueprints.

- Building Codes and Regulations Enforcement: Foundation models can help assimilate information from building codes and regulations (mostly textual data) for better building design and operation.

By leveraging foundation models in smart buildings, various tasks can be efficiently performed, enabling better automation, optimization, and management of building systems, leading to enhanced occupant comfort, and energy efficiency.

Framework for Unseen Data Realization

In critical systems such as smart buildings and healthcare, many situations occur very rarely, and cannot be emulated otherwise because of practical and ethical challenges. However, since ML systems are data hungry, they require instantiations of those situations to train themselves properly. In this case, extrapolation to generate unseen data can be very useful. Hence, an area of future research can be to develop a generative modeling framework, which provides functionalities to manipulate attributes of the original input space to generate desired unseen synthetic samples, and at the same time, populate particular low-frequency classes of a dataset with synthetic samples for data balancing.

For generating data samples with desired qualities, flow and GAN-based generative models that offer a rich latent space can be used, where the manipulation of features can be done [229, 230]. For example, synthetic human behaviors that alleviate the need for a large number of human subjects in thermal comfort studies can be generated using above approach. An experiment might have an overweight individual with normal activity levels and a normal weight individual with high activity levels. Using the limited information about their thermal comfort behaviors, above proposed method can generate behaviors of an overweight individual with high activity levels, which is not originally present among the subjects (Fig 9.2). The behaviors will closely imitate how an actual overweight individual with high activity levels would have had their thermal comfort signature. Above synthetic data generation can be adopted for other smart infrastructure applications such as occupancy detection (how rooms
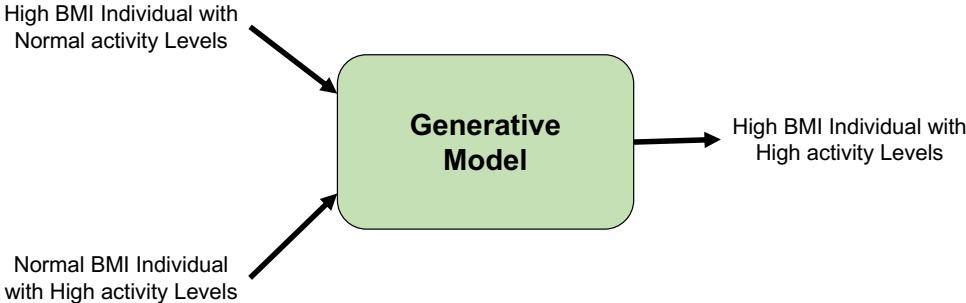
Figure 9.2: Illustration of proposed methodology with example showing unseen thermal comfort signature generation.

of different types be occupied during different times of the day), activity patterns for building control (how different occupants engage with the building) etc.

Bibliography

[1] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. *CoRR*, abs/1605.08803, 2016.

[2] U.S. Department of Energy—Energy Information Administration. Annual Energy Outlook 2020. Table 18. Energy-Related Carbon Dioxide Emissions by Sector and Source, 2020. `https://www.eia.gov/outlooks/aeo`.

[3] Ioannis Konstantakopoulos. *Statistical Learning Towards Gamification in Human-Centric Cyber-Physical Systems*. PhD thesis, EECS Department, University of California, Berkeley, Nov 2018.

[4] Elham Delzendeh, Song Wu, Angela Lee, and Ying Zhou. The impact of occupants' behaviours on building energy analysis: A research review. *Renewable and Sustainable Energy Reviews*, 80:1061 – 1071, 2017.

[5] Bing Dong, Zheng O'Neill, and Zhengwei Li. A BIM-enabled information infrastructure for building energy Fault Detection and Diagnostics. *Automation in Construction*, 44:197–211, 2014.

[6] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *Artificial Neural Networks and Machine Learning–ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part IV*, pages 703–716. Springer, 2019.

[7] Poul O Fanger. Thermal comfort. Analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering.*, 1970.

[8] Shichao Liu, Stefano Schiavon, Hari Prasanna Das, Ming Jin, and Costas J. Spanos. Personal thermal comfort models with wearable sensors. *Building and Environment*, 162:106281, 2019.

[9] Joyce Kim, Stefano Schiavon, and Gail Brager. Personal comfort models–A new paradigm in thermal comfort for occupant-centric environmental control. *Building and Environment*, 132:114–124, 2018.

[10] Xiang Zhou, Ling Xu, Jingsi Zhang, Bing Niu, Maohui Luo, Guangya Zhou, and Xu Zhang. Data-driven thermal comfort model via support vector machine algorithms: Insights from ASHRAE RP-884 database. *Energy and Buildings*, 211:109795, 2020.

[11] Qian Chai, Huiqin Wang, Yongchao Zhai, and Liu Yang. Using machine learning algorithms to predict occupants' thermal comfort in naturally ventilated residential buildings. *Energy and Buildings*, 217:109937, 2020.

[12] Fadi Alsaleem, Mehari K Tesfay, Mostafa Rafaie, Kevin Sinkar, Dhaman Besarla, and Parthiban Arunasalam. An IoT framework for modeling and controlling thermal comfort in buildings. *Frontiers in Built Environment*, 6:87, 2020.

[13] Tanaya Chaudhuri, Deqing Zhai, Yeng Chai Soh, Hua Li, and Lihua Xie. Thermal comfort prediction using normalized skin temperature in a uniform built environment. *Energy and Buildings*, 159:426–440, 2018.

[14] Kuixing Liu, Ting Nie, Wei Liu, Yiqing Liu, and Dayi Lai. A machine learning approach to predict outdoor thermal comfort using local skin temperatures. *Sustainable Cities and Society*, 59:102216, 2020.

[15] Ilaria Pigliautile, Sara Casaccia, Nicole Morresi, Marco Arnesano, Anna Laura Pisello, and Gian Marco Revel. Assessing occupants' personal attributes in relation to human perception of environmental comfort: Measurement procedure and data analysis. *Building and Environment*, 177:106901, 2020.

[16] Toby Cheung, Lindsay T Graham, and Stefano Schiavon. Impacts of life satisfaction, job satisfaction and the Big Five personality traits on satisfaction with the indoor environment. *Building and Environment*, 212:108783, 2022.

[17] Siliang Lu, Weilong Wang, Chaochao Lin, and Erica Cochran Hameen. Data-driven simulation of a thermal comfort-based temperature set-point control with ASHRAE RP884. *Building and Environment*, 156:137–146, 2019.

[18] Jeehee Lee and Youngjib Ham. Physiological sensing-driven personal thermal comfort modelling in consideration of human activity variations. *Building Research & Information*, 49(5):512–524, 2021.

[19] Hari Prasanna Das, Stefano Schiavon, and Costas J Spanos. Unsupervised personal thermal comfort prediction via adversarial domain adaptation. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 230–231, 2021.

[20] Deqing Zhai and Yeng Chai Soh. Balancing indoor thermal comfort and energy consumption of ACMV systems via sparse swarm algorithms in optimizations. *Energy and Buildings*, 149:1–15, 2017.

[21] Aniruddh Chennapragada, Divya Periyakoil, Hari Prasanna Das, and Costas J Spanos. Time series-based deep learning model for personal thermal comfort prediction. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, pages 552–555, 2022.

[22] Toby CT Cheung, Stefano Schiavon, Elliott T Gall, Ming Jin, and William W Nazaroff. Longitudinal assessment of thermal and perceived air quality acceptability in relation to temperature, humidity, and CO2 exposure in Singapore. *Building and Environment*, 115:80–90, 2017.

[23] Joyce Kim, Yuxun Zhou, Stefano Schiavon, Paul Raftery, and Gail Brager. Personal comfort models: Predicting individuals' thermal preference using occupant heating and cooling behavior and machine learning. *Building and Environment*, 129:96–106, 2018.

[24] Shichao Liu. Personal thermal comfort models based on physiological parameters measured by wearable sensors. *Windsor Conference*, 2018.

[25] Nivethitha Somu, Anirudh Sriram, Anupama Kowli, and Krithi Ramamritham. A hybrid deep transfer learning strategy for thermal comfort prediction in buildings. *Building and Environment*, 204:108133, 2021.

[26] Hari Prasanna Das and C. Spanos. Improved Dequantization and Normalization Methods for Tabular Data Pre-Processing in Smart Buildings. In *ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)*, 2022.

[27] Weixin Huang and Hao Zheng. Architectural drawings recognition and generation through machine learning. In *ACADIA*, 2018.

[28] Nelson Nauata, Kai-Hung Chang, Chin-Yi Cheng, Greg Mori, and Yasutaka Furukawa. House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In *European Conference on Computer Vision*, pages 162–177. Springer, 2020.

[29] Vincent JL Gan, HK Wong, Kam Tim Tse, Jack CP Cheng, Irene MC Lo, and Chun Man Chan. Simulation-based evolutionary optimization for energy-efficient layout plan design of high-rise residential buildings. *Journal of cleaner production*, 231:1375–1388, 2019.

[30] Dean Allemang, Jim Hendler, and Fabien Gandon. Expert modeling in OWL, 2011.

[31] Bryan Eisenhower, Zheng O'Neill, Satish Narayanan, Vladimir A Fonoberov, and Igor Mezić. A methodology for meta-model based optimization in building energy models. *Energy and Buildings*, 47:292–301, 2012.

[32] Facundo Bre, Nadia Roman, and Víctor D Fachinotti. An efficient metamodel-based method to carry out multi-objective building performance optimizations. *Energy and buildings*, 206:109576, 2020.

[33] Fisayo Caleb Sangogboye, Ruoxi Jia, Tianzhen Hong, Costas Spanos, and Mikkel Baun Kjærgaard. A framework for privacy-preserving data publishing with enhanced utility for cyber-physical systems. *ACM Transactions on Sensor Networks (TOSN)*, 14(3-4):1–22, 2018.

[34] Ruoxi Jia. *Accountable Data Fusion and Privacy Preservation Techniques in Cyber-Physical Systems*. University of California, Berkeley, 2018.

[35] Dana-Mihaela Petroșanu, George Caruțașu, Nicoleta Luminița Caruțașu, and Alexandru Pirjan. A review of the recent developments in integrating machine learning models with sensor devices in the smart buildings sector with a view to attaining enhanced sensing, energy efficiency, and optimal building management. *Energies*, 12(24):4745, 2019.

[36] Seungwoo Lee, Yohan Chon, Yunjong Kim, Rhan Ha, and Hojung Cha. Occupancy prediction algorithms for thermostat control systems using mobile devices. *IEEE Transactions on Smart Grid*, 4(3):1332–1340, 2013.

[37] Han Zou, Yuxun Zhou, Jianfei Yang, and Costas J Spanos. Unsupervised WiFi-enabled IoT device-user association for personalized location-based service. *IEEE Internet of Things Journal*, 6(1):1238–1245, 2018.

[38] Han Zou, Hao Jiang, Yiwen Luo, Jianjie Zhu, Xiaoxuan Lu, and Lihua Xie. Bluedetect: An ibeacon-enabled scheme for accurate and energy-efficient indoor-outdoor detection and seamless location-based service. *Sensors*, 16(2):268, 2016.

[39] Han Zou, Yuxun Zhou, Hao Jiang, Szu-Cheng Chien, Lihua Xie, and Costas J Spanos. WinLight: A WiFi-based occupancy-driven lighting control system for smart building. *Energy and Buildings*, 158:924–938, 2018.

[40] Avgoustinos Filippoupolitis, William Oliff, and George Loukas. Bluetooth low energy based occupancy detection for emergency management. In *2016 15th international conference on ubiquitous computing and communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*, pages 31–38. IEEE, 2016.

[41] Marek Kraft, Przemysław Aszkowski, Dominik Pieczyński, and Michał Fularz. Low-Cost Thermal Camera-Based Counting Occupancy Meter Facilitating Energy Saving in Smart Buildings. *Energies*, 14(15):4542, 2021.

[42] Yang Zhao, Peter Tu, and Ming-Ching Chang. Occupancy sensing and activity recognition with cameras and wireless sensors. In *Proceedings of the 2nd Workshop on Data Acquisition To Analysis*, pages 1–6, 2019.

[43] Giovanni Diraco, Alessandro Leone, and Pietro Siciliano. People occupancy detection and profiling with 3D depth sensors for building energy management. *Energy and Buildings*, 92:246–266, 2015.

[44] Larry J Brackney, Anthony R Florita, Alex C Swindler, Luigi Gentile Polese, and George A Brunemann. Design and performance of an image processing occupancy sensor. In *Proceedings: The Second International Conference on Building Energy and Environment 2012987 Topic 10. Intelligent buildings and advanced control techniques*. Citeseer, 2012.

[45] Han Zou, Hari Prasanna Das, Jianfei Yang, Yuxun Zhou, and Costas Spanos. Machine Learning empowered Occupancy Sensing for Smart Buildings. *Climate Change + AI Workshop, International Conference on Machine Learning (ICML)*, 2019.

[46] Han Zou, Yuxun Zhou, Jianfei Yang, and Costas J Spanos. Device-free occupancy detection and crowd counting in smart buildings with WiFi-enabled IoT. *Energy and Buildings*, 174:309–322, 2018.

[47] Han Zou, Yuxun Zhou, Jianfei Yang, Weixi Gu, Lihua Xie, and Costas Spanos. Freedetector: Device-free occupancy detection with commodity wifi. In *2017 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*, pages 1–5. IEEE, 2017.

[48] Han Zou, Yuxun Zhou, Jianfei Yang, Huihan Liu, Hari Prasanna Das, and Costas J Spanos. Consensus Adversarial Domain Adaptation. In *AAAI Conference on Artificial Intelligence 2019*, 2019.

[49] MS Zuraimi, A Pantazaras, KA Chaturvedi, JJ Yang, KW Tham, and SE Lee. Predicting occupancy counts using physical and statistical Co2-based modeling methodologies. *Building and Environment*, 123:517–528, 2017.

[50] Krzysztof Arendt, Aslak Johansen, Bo Nørregaard Jørgensen, Mikkel Baun Kjærgaard, Claudio Giovanni Mattera, Fisayo Caleb Sangogboye, Jens Hjort Schwee, and Christian T Veje. Room-level occupant counts, airflow and co2 data from an office building. In *Proceedings of the First Workshop on Data Acquisition To Analysis*, pages 13–14, 2018.

[51] Haolia Rahman and Hwataik Han. Occupancy estimation based on indoor CO2 concentration: comparison of neural network and bayesian methods. *International Journal of Air-Conditioning and Refrigeration*, 25(03):1750021, 2017.

[52] Shuangyu Wei, Paige Wenbin Tien, Tin Wai Chow, Yupeng Wu, and John Kaiser Calautit. Deep learning and computer vision based occupancy CO2 level prediction for demand-controlled ventilation (DCV). *Journal of Building Engineering*, 56:104715, 2022.

[53] Han Zou, Jianfei Yang, Hari Prasanna Das, Huihan Liu, Yuxun Zhou, and Costas J Spanos. Wifi and vision multimodal learning for accurate and robust device-free human activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[54] Veronika Földváry Ličina, Toby Cheung, Hui Zhang, Richard De Dear, Thomas Parkinson, Edward Arens, Chungyoon Chun, Stefano Schiavon, Maohui Luo, Gail Brager, et al. Development of the ASHRAE global thermal comfort database II. *Building and Environment*, 142:502–512, 2018.

[55] Larissa Arakawa Martins, Veronica Soebarto, and Terence Williamson. A systematic review of personal thermal comfort models. *Building and Environment*, 207:108502, 2022.

[56] Camila González, Karol Gotkowski, Moritz Fuchs, Andreas Bucher, Armin Dadras, Ricarda Fischbach, Isabel Jasmin Kaltenborn, and Anirban Mukhopadhyay. Distance-based detection of out-of-distribution silent failures for Covid-19 lung lesion segmentation. *Medical Image Analysis*, 82:102596, 2022.

[57] Hayden Gunraj, Linda Wang, and Alexander Wong. COVIDNet-CT: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest CT Images, 2020.

[58] Varun Kompella, Roberto Capobianco, Stacy Jong, Jonathan Browne, Spencer Fox, Lauren Meyers, Peter Wurman, and Peter Stone. Reinforcement Learning for Optimization of COVID-19 Mitigation policies, 2020.

[59] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[60] Matias Quintana, Stefano Schiavon, Kwok Wai Tham, and Clayton Miller. Balancing thermal comfort datasets: We GAN, but should we? In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 120–129, 2020.

[61] Hiroki Yoshikawa, Akira Uchiyama, and Teruo Higashino. Data balancing for thermal comfort datasets using conditional wasserstein GAN with a weighted loss function. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 264–267, 2021.

[62] Hari Prasanna Das, Ryan Tran, Japjot Singh, Xiangyu Yue, Geoff Tison, Alberto Sangiovanni-Vincentelli, and Costas J Spanos. Conditional Synthetic Data Generation for Robust Machine Learning Applications with Limited Pandemic Data. In *arXiv preprint arXiv:2109.06486*, 2021.

[63] Hari Prasanna Das and Costas J. Spanos. Conditional Synthetic Data Generation for Personal Thermal Comfort Models, 2022.

[64] David Ormandy and Véronique Ezratty. Health and thermal comfort: From WHO guidance to housing strategies. *Energy Policy*, 49:116–121, 2012.

[65] K Pantavou, G Theoharatos, A Mavrakis, and M Santamouris. Evaluating thermal comfort conditions and health responses during an extremely hot summer in Athens. *Building and Environment*, 46(2):339–344, 2011.

[66] Takashi Akimoto, Shin-ichi Tanabe, Takashi Yanai, and Masato Sasaki. Thermal comfort and productivity-Evaluation of workplace environment in a task conditioned office. *Building and environment*, 45(1):45–50, 2010.

[67] Mark J Mendell and Garvin A Heath. Do indoor pollutants and thermal conditions in schools influence student performance? A critical review of the literature. *Indoor air*, 15(1):27–52, 2005.

[68] Li Lan, Zhiwei Lian, and Li Pan. The effects of air temperature on office workers' well-being, workload and productivity-evaluated with subjective ratings. *Applied ergonomics*, 42(1):29–36, 2010.

[69] American National Standards Institute. *Thermal environmental conditions for human occupancy*, volume 55. American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2004.

[70] EN Cen. 15251-2007, criteria for the indoor environment including thermal, indoor air quality, light and noise. *Brussels: European Committee for Standardization*, 2007.

[71] ISO7730 ISO. 7730: Ergonomics of the thermal environment Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria. *Management*, 3(605):e615, 2005.

[72] Richard De Dear and Gail Schiller Brager. Developing an adaptive model of thermal comfort and preference. 1998.

[73] George Havenith, Ingvar Holmér, and Ken Parsons. Personal factors in thermal comfort assessment: clothing properties and metabolic heat production. *Energy and buildings*, 34(6):581–591, 2002.

[74] Toby Cheung, Stefano Schiavon, Thomas Parkinson, Peixian Li, and Gail Brager. Analysis of the accuracy on PMV–PPD model using the ASHRAE Global Thermal Comfort Database II. *Building and Environment*, 153:205–217, 2019.

[75] Da Li, Carol C Menassa, and Vineet R Kamat. Personalized human comfort in indoor building environments under diverse conditioning modes. *Building and Environment*, 126:304–317, 2017.

[76] Md Shajalal, Milad Bohlouli, Hari Prasanna Das, Alexander Boden, and Gunnar Stevens. Focus on what matters: improved feature selection techniques for personal thermal comfort modelling. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 496–499, 2022.

[77] David Daum, Frédéric Haldi, and Nicolas Morel. A personalized measure of thermal comfort for building controls. *Building and Environment*, 46(1):3–11, 2011.

[78] Ali Ghahramani, Chao Tang, and Burcin Becerik-Gerber. An online learning approach for quantifying personalized thermal comfort via adaptive stochastic modeling. *Building and Environment*, 92:86–96, 2015.

[79] Brent Huchuk, William O'Brien, and Scott Sanner. A longitudinal study of thermostat behaviors based on climate, seasonal, and energy price considerations using connected thermostat data. *Building and Environment*, 139:199–210, 2018.

[80] Wilmer Pasut, Hui Zhang, Ed Arens, and Yongchao Zhai. Energy-efficient comfort with a heated/cooled chair: Results from human subject tests. *Building and Environment*, 84:10–21, 2015.

[81] Danni Wang, Hui Zhang, Edward Arens, and Charlie Huizenga. Observations of upper-extremity skin temperature and corresponding overall-body thermal sensations and comfort. *Building and Environment*, 42(12):3933–3943, 2007.

[82] Kizito N Nkurikiyeyezu, Yuta Suzuki, Yoshito Tobe, Guillaume F Lopez, and Kiyoshi Itao. Heart rate variability as an indicator of thermal comfort state. In *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1510–1512. IEEE, 2017.

[83] Fan Zhang, Shamila Haddad, Bahareh Nakisa, Mohammad Naim Rastgoo, Christhina Candido, Dian Tjondronegoro, and Richard de Dear. The effects of higher temperature setpoints during summer on office workers' cognitive load and thermal comfort. *Building and environment*, 123:176–188, 2017.

[84] A Pharo Gagge, JAJ Stolwijk, and JD Hardy. Comfort and thermal sensations and associated physiological responses at various ambient temperatures. *Environmental research*, 1(1):1–20, 1967.

[85] Megan P Rothney, Emily V Schaefer, Megan M Neumann, Leena Choi, and Kong Y Chen. Validity of physical activity intensity predictions by ActiGraph, Actical, and RT3 accelerometers. *Obesity*, 16(8):1946–1952, 2008.

[86] Soo Young Sim, Myung Jun Koh, Kwang Min Joo, Seungwoo Noh, Sangyun Park, Youn Ho Kim, and Kwang Suk Park. Estimation of thermal sensation based on wrist skin temperatures. *Sensors*, 16(4):420, 2016.

[87] Moatassem Abdallah, Caroline Clevenger, Tam Vu, and Anh Nguyen. Sensing occupant comfort using wearable technologies. In *Construction Research Congress 2016*, pages 940–950, 2016.

[88] Foster J Provost, Tom Fawcett, Ron Kohavi, et al. The case against accuracy estimation for comparing induction algorithms. In *ICML*, volume 98, pages 445–453, 1998.

[89] Ana Uzelac, Nenad Gligoric, and Srdjan Krco. A comprehensive study of parameters in physical environment that impact students' focus during lecture using Internet of Things. *Computers in Human Behavior*, 53:427–434, 2015.

[90] Nan Ma, Liang Chen, Jian Hu, Paris Perdikaris, and William W. Braham. Adaptive behavior and different thermal experiences of real people: A Bayesian neural network approach to thermal preference prediction and classification, Apr 2021.

[91] Andrei Claudiu Cosma and Rahul Simha. Thermal comfort modeling in transient conditions using real-time local body temperature extraction with a thermographic camera. *Building and Environment*, 143:36–47, 2018.

[92] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[93] Ioannis C Konstantakopoulos, Andrew R Barkan, Shiying He, Tanya Veeravalli, Huihan Liu, and Costas Spanos. A deep learning and gamification approach to improving human-building interaction and energy efficiency in smart infrastructure. *Applied energy*, 237:810–821, 2019.

[94] S. Liu, M. Jin, H. Das, C. Spanos, and S. Schiavon. Personal thermal comfort models based on physiological parameters measured by wearable sensors. *Proceedings of the Windsor Conference*, pages 431–441, 2018.

[95] Ioannis Konstantakopoulos, Kristy Hamilton, Yashaswini Murthy, Tanya Veeravalli, Costas Spanos, and Roy Dong. smartSDH: An Experimental Study of Mechanism-Based Building Control. *IEEE Systems Journal*, 16(4):6289–6299, 2022.

[96] Ioannis C Konstantakopoulos, Kristy A Hamilton, Tanya Veeravalli, Costas Spanos, and Roy Dong. smartSDH: A Mechanism Design Approach to Building Control. *arXiv preprint arXiv:2001.02807*, 2020.

[97] J. Kwac, C. Tan, N. Sintov, J. Flora, and R. Rajagopal. Utility customer segmentation based on smart meter data: Empirical study. In *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 720–725, Oct 2013.

[98] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015.

[99]   Y. Zuo, G. Yu, and H. W. Ressom. Integrating prior biological knowledge and graphical LASSO for network inference. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1543–1547, Nov 2015.

[100] N. Souly and M. Shah. Scene Labeling Using Sparse Precision Matrix. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3650–3658, June 2016.

[101] C.W.J. Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and Control*, 2:329 – 352, 1980.

[102] Hari Prasanna Das, Ioannis Konstantakopoulos, Aummul Baneen Manasawala, Tanya Veeravalli, Huihan Liu, and Costas J Spanos. Do Occupants in a Building exhibit patterns in Energy Consumption? Analyzing Clusters in Energy Social Games. 2020.

[103] Hari Prasanna Das, Ioannis C Konstantakopoulos, Aummul Baneen Manasawala, Tanya Veeravalli, Huihan Liu, and Costas J Spanos. Segmentation analysis in human centric cyber-physical systems using graphical lasso. *arXiv preprint arXiv:1810.10533*, 2018.

[104] B. Ai, Z. Fan, and R. X. Gao. Occupancy estimation for smart buildings by an auto-regressive hidden Markov model. In *2014 American Control Conference*, pages 2234–2239, June 2014.

[105] Erik Knol and Peter W de Vries. EnerCities—A Serious Game to Stimulate Sustainability and Energy Conservation: Preliminary Results. *eLearning Papers*, 2011.

[106] M. Roozbehani, M. Dahleh, and S. Mitter. Dynamic Pricing and Stabilization of Supply and Demand in Modern Electric Power Grids. In *1st IEEE Int. Conf. Smart Grid Communications*, pages 543–548, oct. 2010.

[107] Ioannis C Konstantakopoulos, Lillian J Ratliff, Ming Jin, Costas J Spanos, and S Shankar Sastry. Inverse modeling of non-cooperative agents via mixture of utilities. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 6327–6334. IEEE, 2016.

[108] Ioannis C Konstantakopoulos, Lillian J Ratliff, Ming Jin, and Costas J Spanos. Leveraging correlations in utility learning. In *American Control Conference (ACC), 2017*, pages 5249–5256. IEEE, 2017.

[109] Ioannis C Konstantakopoulos, Lillian J Ratliff, Ming Jin, S Shankar Sastry, and Costas J Spanos. A robust utility learning framework via inverse optimization. *IEEE Transactions on Control Systems Technology*, 26(3):954–970, 2018.

[110] S. Li, K. Deng, and M. Zhou. Social incentive policies to engage commercial building occupants in demand response. In *IEEE Inter. Conf. Automation Science and Engineering*, pages 407–412, Aug 2014.

[111] Lillian J Ratliff, Ming Jin, Ioannis C Konstantakopoulos, Costas Spanos, and S Shankar Sastry. Social game for building energy efficiency: Incentive design. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1011–1018. IEEE, 2014.

[112] David Hallac, Youngsuk Park, Stephen P. Boyd, and Jure Leskovec. Network Inference via the Time-Varying Graphical Lasso. *CoRR*, abs/1703.01958, 2017.

[113] Song Zan Chiou-Wei, Ching-Fu Chen, and Zhen Zhu. Economic growth and energy consumption revisited evidence from linear and nonlinear Granger causality. *Energy Economics*, 30(6):3063–3076, 2008.

[114] Hari Prasanna Das. Graphical Lasso based Cluster Analysis in Energy-Game Theoretic Frameworks. 2021.

[115] Hari Prasanna Das, Ioannis C Konstantakopoulos, Aummul Baneen Manasawala, Tanya Veeravalli, Huihan Liu, and Costas J Spanos. A novel graphical lasso based approach towards segmentation analysis in energy game-theoretic frameworks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1702–1709. IEEE, 2019.

[116] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[117] P. Robert and Y. Escoufier. A Unifying Tool for Linear Multivariate Statistical Methods: The RV- Coefficient. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 25(3):257–265, 1976.

[118] Lee Schipper and Michael Grubb. On the rebound? Feedback between energy intensities and energy uses in IEA countries. *Energy Policy*, 28(6–7):367–388, 2000.

[119] R. Jia, I. C. Konstantakopoulos, B. Li, and C. Spanos. Poisoning Attacks on Data-Driven Utility Learning in Games. In *2018 Annual American Control Conference (ACC)*, pages 5774–5780, June 2018.

[120] Lillian J Ratliff and Tanner Fiez. Adaptive Incentive Design. *arXiv preprint arXiv:1806.05749*, 2018.

[121] Diogo Gomes and João Saúde. A mean-field game approach to price formation in electricity markets. *arXiv preprint arXiv:1807.07088*, 2018.

[122] Ming Jin, Ruoxi Jia, Hari Prasanna Das, Wei Feng, and Costas Spanos. BISCUIT: Building Intelligent System CUstomer Investment Tool. In *10th International Conference on Applied Energy (ICAE)*, 2018.

[123] Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed Impact of Fair Machine Learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3150–3158. PMLR, 2018.

[124] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[125] Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[126] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 854–863. JMLR. org, 2017.

[127] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. *CoRR*, abs/1701.05517, 2017.

[128] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[129] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

[130] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. *arXiv preprint arXiv:1712.09763*, 2017.

[131] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[132] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[133] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

[134] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[135] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. *arXiv preprint arXiv:1811.00995*, 2018.

[136] Andrei Atanov, Alexandra Volokhova, Arsenii Ashukha, Ivan Sosnovik, and Dmitry Vetrov. Semi-Conditional Normalizing Flows for Semi-Supervised Learning. *arXiv preprint arXiv:1905.00505*, 2019.

[137] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. VideoFlow: A flow-based generative model for video. *arXiv preprint arXiv:1903.01434*, 2019.

[138] Hari Prasanna Das, Pieter Abbeel, and Costas J Spanos. Likelihood Contribution based Multi-scale Architecture for Generative Flows. *arXiv preprint arXiv:1908.01686*, 2019.

[139] Hari Prasanna Das, Pieter Abbeel, and Costas J Spanos. Dimensionality reduction flows. *arXiv preprint arXiv:1908.01686*, pages 1–10, 2019.

[140] Benigno Uria, Iain Murray, and Hugo Larochelle. RNADE: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*, pages 2175–2183, 2013.

[141] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. *arXiv preprint arXiv:1902.00275*, 2019.

[142] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[143] Ricky TQ Chen, Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. Residual Flows for Invertible Generative Modeling. *arXiv preprint arXiv:1906.02735*, 2019.

[144] Will Grathwohl, Ricky TQ Chen, Jesse Betterncourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[145] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-Supervised Learning with Normalizing Flows. In *Workshop on Invertible Neural Nets and Normalizing Flows, International Conference on Machine Learning*, 2019.

[146] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Cubic-Spline Flows. *arXiv preprint arXiv:1906.02145*, 2019.

[147] Emiel Hoogeboom, Rianne van den Berg, and Max Welling. Emerging convolutions for generative normalizing flows. *arXiv preprint arXiv:1901.11137*, 2019.

[148] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[149] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet Large Scale Visual Recognition Challenge. *CoRR*, abs/1409.0575, 2014.

[150] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[151] Xuezhe Ma, Xiang Kong, Shanghang Zhang, and Eduard H Hovy. Decoupling Global and Local Representations via Invertible Generative Flows. In *International Conference on Learning Representations*, 2021.

[152] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP, 2017.

[153] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[154] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[155] Yuxun Zhou and Costas J Spanos. Causal meets submodular: Subset selection with directed information. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2657–2665. Citeseer, 2016.

[156] Linda Wang, Zhong Qiu Lin, and Alexander Wong. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Scientific Reports*, 10(1):19549, Nov 2020.

[157] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[158] Aditya Mishra. Metrics to Evaluate your Machine Learning Algorithm. 2018.

[159] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, 2018.

[160] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions, 2014.

[161] Xuezhe Ma, Xiang Kong, Shanghang Zhang, and Eduard Hovy. Decoupling Global and Local Representations from/for Image Generation, 2020.

[162] Amirata Ghorbani, Vivek Natarajan, David Coz, and Yuan Liu. DermGAN: Synthetic Generation of Clinical Skin Images with Pathology, 2019.

[163] Timo Kohlberger, Yun Liu, Melissa Moran, Po-Hsuan Cameron Chen, Trissia Brown, Jason D Hipp, Craig H Mermel, and Martin C Stumpe. Whole-slide image focus quality: Automatic assessment and impact on ai cancer detection. *Journal of pathology informatics*, 10, 2019.

[164] Tianyu Han, Sven Nebelung, Christoph Haarburger, Nicolas Horst, Sebastian Reinartz, Dorit Merhof, Fabian Kiessling, Volkmar Schulz, and Daniel Truhn. Breaking Medical Data Sharing Boundaries by Employing Artificial Radiographs. *bioRxiv*, 2019.

[165] Nayana Bannur, Vishwa Shah, Alpan Raval, and Jerome White. Synthetic Data Generation for Improved covid-19 Epidemic Forecasting. *medRxiv*, 2020.

[166] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro. CovidGAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection. *IEEE Access*, 8:91916–91923, 2020.

[167] Yifan Jiang, Han Chen, Murray Loew, and Hanseok Ko. Covid-19 ct image synthesis with a conditional generative adversarial network. *IEEE Journal of Biomedical and Health Informatics*, 25(2):441–452, 2020.

[168] Rui Liu, Yu Liu, Xinyu Gong, Xiaogang Wang, and Hongsheng Li. Conditional Adversarial Generative Flow for Controllable Image Synthesis, 2019.

[169] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved Techniques for Training GANs, 2016.

[170] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design, 2019.

[171] Hari Prasanna Das, Ryan Tran, Japjot Singh, Yu-Wen Lin, and Costas J. Spanos. CDCGen: Cross-Domain Conditional Generation via Normalizing Flows and Adversarial Training, 2021.

[172] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes, 2014.

[173] Saeid Motiian, Quinn Jones, Seyed Mehdi Iranmanesh, and Gianfranco Doretto. Few-Shot Adversarial Domain Adaptation, 2017.

[174] Takeshi Teshima, Issei Sato, and Masashi Sugiyama. Few-shot domain adaptation by causal mechanism transfer. In *International Conference on Machine Learning*, pages 9458–9469. PMLR, 2020.

[175] An Zhao, Mingyu Ding, Zhiwu Lu, Tao Xiang, Yulei Niu, Jiechao Guan, and Ji-Rong Wen. Domain-adaptive few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1390–1399, 2021.

[176] Kushagra Mahajan, Monika Sharma, and Lovekesh Vig. Meta-dermdiagnosis: Few-shot skin disease identification using meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 730–731, 2020.

[177] Deepta Rajan, Jayaraman J Thiagarajan, Alexandros Karargyris, and Satyananda Kashyap. Self-training with improved regularization for sample-efficient chest x-ray classification. In *Medical Imaging 2021: Computer-Aided Diagnosis*, volume 11597, page 115971S. International Society for Optics and Photonics, 2021.

[178] Shruti Jadon. COVID-19 detection from scarce chest x-ray image data using few-shot deep learning approach. In *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications*, volume 11601, page 116010X. International Society for Optics and Photonics, 2021.

[179] Hari Prasanna Das and Costas J Spanos. Synthetic personal thermal comfort data generation. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 280–281, 2022.

[180] Navigant Research Report. Intelligent Building Technologies for Sustainability.

[181] Jack Ngarambe, Geun Young Yun, and Mat Santamouris. The use of artificial intelligence (AI) methods in the prediction of thermal comfort in buildings: Energy implications of AI-based thermal comfort controls. *Energy and Buildings*, 211:109807, 2020.

[182] Mohammad Taleghani, Martin Tenpierik, Stanley Kurvers, and Andy Van Den Dobbel-steen. A review into thermal comfort in buildings. *Renewable and Sustainable Energy Reviews*, 26:201–215, 2013.

[183] Maohui Luo, Zhe Wang, Kevin Ke, Bin Cao, Yongchao Zhai, and Xiang Zhou. Human metabolic rate and thermal comfort in buildings: The problem and challenge. *Building and Environment*, 131:44–52, 2018.

[184] Tanaya Chaudhuri, Yeng Chai Soh, Hua Li, and Lihua Xie. Machine learning based prediction of thermal comfort in buildings of equatorial Singapore. In *2017 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC)*, pages 72–77. IEEE, 2017.

[185] Divya Periyakoil, Hari Prasanna Das, and Costas J Spanos. Understanding Distributions of Environmental Parameters for Thermal Comfort Study in Singapore. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, pages 461–465, 2020.

[186] Divya Periyakoil, Hari Prasanna Das, Clayton Miller, Costas J Spanos, and Ndola Prata. Environmental Exposures in Singapore Schools: An Ecological Study. *International journal of environmental research and public health*, 18(4):1843, 2021.

[187] Prageeth Jayathissa, Matias Quintana, Mahmoud Abdelrahman, and Clayton Miller. Humans-as-a-sensor for buildings—intensive longitudinal indoor comfort models. *Buildings*, 10(10):174, 2020.

[188] Kailai Sun, Qianchuan Zhao, and Jianhong Zou. A review of building occupancy measurement systems. *Energy and Buildings*, 216:109965, 2020.

[189] Ioannis C Konstantakopoulos, Hari Prasanna Das, Andrew R Barkan, Shiying He, Tanya Veeravalli, Huihan Liu, Aummul Baneen Manasawala, Yu-Wen Lin, and Costas J Spanos. Design, benchmarking and explainability analysis of a game-theoretic framework towards energy efficiency in smart infrastructure. *arXiv preprint arXiv:1910.07899*, 2019.

[190] Arash Khalilnejad, Roger H French, and Alexis R Abramson. Data-driven evaluation of HVAC operation and savings in commercial buildings. *Applied Energy*, 278:115505, 2020.

[191] Siavash H Khajavi, Naser Hossein Motlagh, Alireza Jaribion, Liss C Werner, and Jan Holmström. Digital twin: vision, benefits, boundaries, and creation for buildings. *IEEE access*, 7:147406–147419, 2019.

[192] Rasmus Luthander, Annica M Nilsson, Joakim Widén, and Magnus Åberg. Graphical analysis of photovoltaic generation and load matching in buildings: A novel way of studying self-consumption and self-sufficiency. *Applied Energy*, 250:748–759, 2019.

[193] Jie Zhao, Bertrand Lasternas, Khee Poh Lam, Ray Yun, and Vivian Loftness. Occupant behavior and schedule modeling for building energy simulation through office appliance power consumption data mining. *Energy and Buildings*, 82:341–355, 2014.

[194] Rui Tang and Shengwei Wang. Model predictive control for thermal energy storage and thermal comfort optimization of building demand response in smart grids. *Applied Energy*, 242:873–882, 2019.

[195] Ahmet Uğursal and Charles H Culp. The effect of temperature, metabolic rate and dynamic localized airflow on thermal comfort. *Applied energy*, 111:64–73, 2013.

[196] Zimu Zheng, Yimin Dai, and Dan Wang. DUET: Towards a portable thermal comfort model. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 51–60, 2019.

[197] Lei Xiong and Ye Yao. Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm. *Building and Environment*, 202:108026, 2021.

[198] Wooyoung Jung and Farrokh Jazizadeh. Comparative assessment of HVAC control strategies using personal thermal comfort and sensitivity models. *Building and Environment*, 158:104–119, 2019.

[199] Zhun Yu, Benjamin CM Fung, Fariborz Haghighat, Hiroshi Yoshino, and Edward Morofsky. A systematic procedure to study the influence of occupant behavior on building energy consumption. *Energy and buildings*, 43(6):1409–1417, 2011.

[200] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[201] Zhe Wang, Hui Zhang, Yingdong He, Maohui Luo, Ziwei Li, Tianzhen Hong, and Borong Lin. Revisiting individual and group differences in thermal comfort based on ASHRAE database. *Energy and Buildings*, 219:110017, 2020.

[202] Tobias Kramer, Veronica Garcia-Hansen, Sara Omrani Vahid M Nik, and Dong Chen. A Machine Learning approach to enhance indoor thermal comfort in a changing climate. In *Journal of Physics: Conference Series*, volume 2042, page 012070. IOP Publishing, 2021.

[203] Andrew Sonta, Thomas R Dougherty, and Rishee K Jain. Data-driven optimization of building layouts for energy efficiency. *Energy and Buildings*, 238:110815, 2021.

[204] Ioannis Antonopoulos, Valentin Robu, Benoit Couraud, and David Flynn. Data-driven modelling of energy demand response behaviour based on a large-scale residential trial. *Energy and AI*, 4:100071, 2021.

[205] Nikolaos Nasios and Adrian G Bors. Variational learning for Gaussian mixture models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(4):849–862, 2006.

[206] Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74:902–924, 2017.

[207] Renzhi Lu, Seung Ho Hong, and Mengmeng Yu. Demand response for home energy management using reinforcement learning and artificial neural network. *IEEE Transactions on Smart Grid*, 10(6):6629–6639, 2019.

[208] Ján Drgoňa, Aaron R Tuor, Vikas Chandan, and Draguna L Vrabie. Physics-constrained deep learning of multi-zone building thermal dynamics. *Energy and Buildings*, 243:110992, 2021.

[209] Hari Prasanna Das, Ryan Tran, Japjot Singh, Yu-Wen Lin, and Costas J Spanos. Unsupervised Cross-Domain Conditional Generation via Normalizing Flows and Adversarial Training. 2022.

[210] Lindsay T Graham, Thomas Parkinson, and Stefano Schiavon. Lessons learned from 20 years of CBE's occupant surveys. *Buildings and Cities*, 2(1), 2021.

[211] Nan Gao, Wei Shao, Mohammad Saiedur Rahaman, Jun Zhai, Klaus David, and Flora D. Salim. Transfer Learning for Thermal Comfort Prediction in Multiple Cities, 2020.

[212] Annamalai Natarajan and Emil Laftchiev. A transfer active learning framework to predict thermal comfort. *International Journal of Prognostics and Health Management*, 10(3), 2019.

[213] Xiangyu Yue, Zangwei Zheng, Hari Prasanna Das, Kurt Keutzer, and Alberto Sangiovanni Vincentelli. Multi-source Few-shot Domain Adaptation. *arXiv preprint arXiv:2109.12391*, 2021.

[214] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, 2017.

[215] Amélie Royer, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole, and Kevin Murphy. Xgan: Unsupervised image-to-image translation for many-to-many mappings. In *Domain Adaptation for Visual Understanding*, pages 33–49. Springer, 2020.

[216] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 3730–3739, 2017.

[217] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[218] Aditya Grover, Christopher Chute, Rui Shu, Zhangjie Cao, and Stefano Ermon. Align-Flow: Cycle Consistent Learning from Multiple Domains via Normalizing Flows. *arXiv preprint arXiv:1905.12892*, 2019.

[219] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[220] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[221] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[222] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017.

[223] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7986–7994, 2018.

[224] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[225] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[226] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[227] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[228] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[229] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[230] Chris Donahue, Zachary C Lipton, Akshay Balsubramani, and Julian McAuley. Semantically decomposing the latent spaces of generative adversarial networks. *arXiv preprint arXiv:1705.07904*, 2017.