Statistical Learning Towards Gamification in Human-Centric Cyber-Physical Systems

by

Ioannis Konstantakopoulos

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 Spanos, Chair
Professor Shankar Sastry
Professor Stefano Schiavon
Professor Alexandra von Meier

Fall 2018

Statistical Learning Towards Gamification in Human-Centric Cyber-Physical Systems

Abstract

Statistical Learning Towards Gamification in Human-Centric Cyber-Physical Systems

by

Ioannis Konstantakopoulos

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Costas Spanos, Chair

This dissertation thesis explores *Human-Centric Cyber-Physical Systems* by simultaneously considering users' behavior/preference and their interaction as strategic agents. We envision smart-building systems in which humans take control, interact, and improve the environment they live in. People's interaction in a cyber-physical system is a core mechanism of the implementation of smart building technology. Adoption of human-centric building services and amenities also leads to improvements in the operational efficiency of cyber-physical systems that are used to control building energy usage. However, human preference in regard to living conditions is usually unknown and heterogeneous in its manifestation as control inputs to a building. Furthermore, the occupants of a building typically lack the independent motivation necessary to contribute to and play a key role in the control of smart building infrastructure. We focus on the development of a generalized gamification abstraction towards enabling strategic interactions among non-cooperative agents in *Human-Centric Cyber-Physical Systems*. The proposed framework enables a humans-in-the-loop strategy using an interface to allow building managers to interact with occupants. This interface is designed for occupants' engagement—integration while it supports learning occupants' preferences over shared or scarce resources in addition to understanding how preferences change as a function of external stimuli such as physical control, time or incentives. Our gamification framework can be used in the design of incentive mechanisms that realign agents' preferences with those of the planner which often represent system-level performance criteria through fair compensation.

In our first approach, we model user interaction as a continuous game between non-cooperative players. Game theoretic analysis often relies on the assumption that the utility function of each agent is known a priori; however, this assumption usually does not hold in many real-world applications. We propose a parametric utility learning framework leveraging inverse optimization techniques and explore vulnerability from adversarial attacks in utility learning and present potential security risks. A generalized robust framework of the proposed learning method is introduced by employing constrained feasible generalized least squares estimations with heteroskedastic inference. We further develop the theoretical formulation

of a new parametric utility learning method that uses a probabilistic interpretation—i.e. a mixture of utilities—of agent utility functions that allows us to account for variations in agents' parameters over time. Furthermore, towards the reduction of the complexity of the advanced learning methods we propose a new method of data-driven modeling of human decision-making by accounting for possible correlations between players and form coalitions between agents.

Advancements in cyber-physical systems lead to the collection of more and more data as a result of users' interactions with cyber-physical systems' sensing/actuation platforms. This enables new ways to improve infrastructure systems and lead to smart-building energy efficiency. Towards modeling users in their engagement and integration in a *Human-Centric Cyber-Physical System*, we characterize their interaction as a sequential discrete game between non-cooperative players. We propose the design and implementation of a large-scale network gamification application with the goal of improving the energy efficiency of a building through the utilization of cutting-edge Internet of Things (IoT) sensors and cyber-physical systems sensing/actuation platforms. Then, by observing human decision-makers and their decision strategies in their operation of building systems, we can apply inverse learning techniques in order to estimate their utility functions. We propose a benchmark utility learning framework that employs robust estimations for classical discrete choice models provided with high dimensional imbalanced data. To improve forecasting performance, we extend the benchmark utility learning scheme by leveraging Deep Learning end-to-end training with Deep bi-directional Recurrent Neural Networks. Most importantly, we use conventional deep variational auto-encoders and recurrent network based adaptation of variational auto-encoders as an approach to create nonlinear manifolds (encoders) that can be used as a generative model of agents' decision-making process.

A series of experimental trials was conducted to generate real-world data, which was then used as the main source of data for our approaches. We apply the proposed methods to data from social game experiments designed to encourage energy efficient behavior among smart building occupants in the Nanyang Technological University (NTU) residential housing and the Center for Research in Energy Systems Transformation (CREST) on the UC Berkeley campus. This differentiates our work from a large portion of other works in the same field that use simulations in lieu of experimental methods.

"I am indebted to my father for living, but to my teacher for living well."
– Alexander the Great

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

I am indebted to Professor Costas Spanos for mentoring and constantly supporting me during my Doctoral studies. Costas was always helpful and patient with every question—problem that I came across through my PhD journey. He believed in me and my motivations for PhD research while providing significant feedback. Undoubtedly, I would not have been where I am today without Costas' help and support.

I would like to thank my Dissertation & Qualification exam committee members. Professor Shankar Sastry provided me amazing support and invaluable guidance. Every single conversation with Shankar is a lifelong advice and his feedback in my research problems was so helpful. Moreover, Professor Alexandra von Meier introducing me several aspects of smart grid technology. Alexandra gave me the opportunity to spend a fantastic semester as Graduate Student Instructor at her class. This was an invaluable teaching experience. Professor Stefano Schiavon for giving me guidance about smart building technology and especially helping me to understand human thermal-comfort. His feedback lead to most of my research ideas for human-building interaction.

I would like to thank all those researchers who worked close with me and helped with the work of this dissertation. Dr. Ming Jin for our work in inverse learning for Nash based games. Ming was always a source of feedback and support for the most important parts of this thesis. Dr. Ruoxi Jia for a fantastic viewpoint of malicious attacks in learning problems and our common efforts in several graduate classes. Professor Lillian Ratliff for her support on game theory models and unique viewpoint on research. Also, thanks to Andrew R. Barkan, Shiying He, Tanya Veeravalli, and Huihan Liu for our great collaboration with the Deep Learning research part of this dissertation.

I would also like to thank the people at CREST, SinBerBEST, Energy Research Institute (ERI@N) at Nanyang Technological University, and EECS for all those wonderful years: Ming Jin, Ruoxi Jia, Jason Poon, Han Zhou, Yuxun Zhou, Hari Prasanna Das, Lucas Spangher, Wendy Lin, Utkarsha Agwan, Zhaoyi Kang, Ying Qiao, Jae Yeon Baek, Li Dan, Chris Hsu, Chris Soyza, Daphne Pakiam, Ramit Anand Sharma, Chayle Batiquin, Geraldine Thoung, Patricia Alvina and Nilesh Y. Jadhav. Special thanks to Chris Hsu, the applications programmer at CREST laboratory, who developed and deployed the web portal applications as well as the social game data pipeline architectures for all real-time experiments. Moreover, the Nanyang Technological University gamification project could not succeed without the important support of our interns at Nanyang Technological University campus: Ying Xuan Chua, Shuen Hwee Yee, Keryn Kan, Cyndi Shin Yi Teh, Shu Yu Tan, and Yu Jie Lee. Our intern team helped in several challenges before and during the deployment of our social game. Undoubtedly, their support was highly important for the experiment and its success.

I am also grateful to several staff members of CREST, EECS, and Berkeley international office for their incredible support: Yovana Gomez, Judy Huang, Shirley Salanio, Patrick Hernan, Yulia Golubovskaya, and Amy Chin-Pokhrel. Yovana was always a joy to work with and very supportive to our requests. Shirley Salanio was constantly helping me as everything to run smoothly. Shirley was always king, happy and ready to help. I will never

forget her smile every time she replied to my questions! Lastly, Amy Chin-Pokhrel was so kind and helpful in my needs as international student. Amy provided feedback and clear paths as to avoid any problem regarding my status as international student.

Special thanks to all my mentors during my time in graduate school: Roy Dong for his advice in algorithmic game theory, Dorsa Sadigh for guiding me towards my Qualification exam preparation, Aaron Bestick and Walid Krichene for their impressive support during my prelim exam preparation. Thanks to all folks in the UC Berkeley & EECS community: Roel Dobbe, Jaime F. Fisac, Cathy Wu, Eric Kim, Vasuki Narasimha, Aummul Baneen Manasawala, and Jon Tamir.

This work was supported by the Republic of Singapores National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore for the SingaporeBerkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. Also, this work was supported by a scholarship of the Alexander S. Onassis Public Benefit Foundation. Specifically, I want to thank Magkel Katerina for her support and guidance regarding Alexander S. Onassis scholarship.

I want to thank my family, both in Greece and U.S.A. Thanks to Christos Konstantakopoulos and Fotini Konstantakopoulou for all their love and support. Christos, my father, was supporting to all of my efforts to pursue graduate studies in the US and move all the way across Atlantic Ocean. He was also helping me constantly in my school years and being my very first math mentor. Fotini, my grandmother, was an inspiring role model as a child and for all of my years as a student. Her emotional support and incredible understanding of the difficulties of such effort was very encouraging. Her motivation towards hard work in every part of her life was for me an inspiration for pursuing my dreams. I want also to thank my parents and sister in law for their help and support during my graduate studies. Giannis Kakalis, Christina Anastasopoulou and Eugenia Kakali were very close with me regardless of the distance and also very supportive. I am also so grateful for my relatives in the US side. I cannot thank more Diamantis Kourkouzelis, Tina Bratis, Nikos Kourkouzelis, Marietta Kourkouzelis, Marlen Bratis, Toula Bratis, Erick Gadala, Denise Dionysia Bratis, and Peter Bratis. I will never forget Diamantis phone call once I had arrived in California wanting to immediately help with my relocation! Diamantis is such a great and kind person. He was always helpful and understanding my problems during my studies. Tina was a joy to talk and discuss about various aspects and issues during my PhD studies. She was amazingly supportive and helpful. I truly have no way to thank Diamantis and Tina for their invaluable presence in our life. Thanks!

Most importantly, I would like to dedicate this thesis to my beautiful wife Georgia Kakali, who moved with me all the way across Atlantic Ocean. Georgia makes every single day so special. She supports me from my first year as an undergraduate student in Greece till the end of my PhD studies. All these years Georgia is so supportive and I shall be forever grateful for her help—love. Georgia thanks for your patience, love, support, feedback and for always encouraging me to pursue my dreams. I am deeply grateful for having you in my life. Lastly, I want to say a huge thanks to Eros, our fluffy cat, who stayed awake almost all the nights with me before any conference or other critical deadline. Eros was a constant

support and a playful break during my studies at UC Berkeley.

Chapter 1

Introduction

Energy consumption of buildings, both residential and commercial, accounts for approximately 40% of all energy usage in the U.S. [125]. In efforts to improve energy efficiency in buildings, researchers and industry leaders have attempted to implement novel control and automation approaches alongside techniques like incentive design and adaptive price adjustment to more effectively regulate energy usage. Another common avenue for the regulation of energy usage in buildings is through their on-site building and facility managers. Typically, building managers are obligated to maintain an energy efficient building due to some standard operating procedure or protocol dictated by governing agents. An ideal smart building infrastructure respects and adequately accommodates occupant preferences involving thermal comfort [88], satisfaction/well-being [47], lighting comfort [9], acoustical quality [154], indoor air quality [168], indoor environmental monitoring [78], privacy [70] and productivity [179], while simultaneously optimizing for energy efficiency in addition to agile connectivity with the grid.

Recently, utility companies have invested in demand response programs that can address improper load forecasting while also helping building managers encourage energy efficiency among building occupants [1, 123]. Typically, the implementation of these programs is enacted on a contract basis between utility providers and the consumers under arranged conditions of demand—usage. The building managers will then be bound by contract to operate according to the agreed upon schedule. However, the conditions of these contracts are static and do not consider dynamic changes in occupant behavior or preferences, which can result in discrepancies in demand/usage expectations. To facilitate the adoption of more dynamic protocols for demand response, we propose a gamification interface that allows building managers to interact with a building's occupants. By leveraging our gamification interface, retailers and utility companies can utilize a wealth of dynamic and temporal data on building energy usage, extending even to occupant usage predictions, in order to customize demand response program approaches to observed or predicted conditions [81, 82]. To illustrate the usefulness of our gamification framework, we can consider its potential for application in the context of Stackelberg game approaches, which are proposed as hierarchical control models towards supply-demand management [188], or hierarchical electricity

market level management [187]. Above all, our gamification interface is designed to support engagement and integration on multiple levels in a *human-centric cyber-physical system*. Human-centric cyber-physical systems can be defined as follows:

Systems or mechanisms that combine computer-based technologies with physical processes to integrate direct human coordination through various interaction modalities.

Thus, through automation and integration of the end-user, smart buildings play an integral role in creating a more sustainable and efficient *smart city*.

1.1 Motivation

The implementation of smart building technology in the form of smart infrastructure applications has great potential to improve sustainability and energy efficiency by leveraging a humans-in-the-loop strategy. Adoption of human-centric building services and amenities also leads to improvements in the operational efficiency of cyber-physical systems that are used to control building. However, human preference with regards to living conditions is usually unknown and heterogeneous in its manifestation as control inputs to a building. Furthermore, the occupants of a building typically lack the independent motivation necessary to contribute to and play a key role in the control of smart building infrastructure. Moreover, true human actions and their integration with sensing/actuation platforms remains unknown to the decision maker tasked with improving operational efficiency.

Most importantly, in a *human-centric cyber-physical system*, humans' actions should not be treated as an isolated part of the overall system or noise around dynamics of such systems. Humans are not an overall evolving external disturbances. In several cases humans are rational intelligent agents with pre-defined optimal or sub-optimal strategies, which a cyber-physical system should take in account. Hence, humans need to be treated as an efficient component of the environment they live in. One crucial motivation is about learning upper & lower level models for human—system interaction and dealing with resulting uncertainty in such environments. *Obviously there is a bi-directional dependence between learned models of the environment and the models that express humans' interaction with them.* A variety of fundamental problems needs to be studied in a human-centric cyber-physical system and especially in a smart-building setting.

What is the best way to engage humans regarding energy efficient behaviors in a smart-building setting? How can we model their interaction? How do we model humans and most importantly the interaction between the human and the building's system? How do humans interact between each other in such an environment and does this interaction affect systems' dynamics? How do we manage big data flowing from human-centric cyber-physical system? What are the best learning techniques for those systems? Are these learning techniques robust and (or) can we address safety in a possible adversarial setting?

Our approach leverages gamification techniques to model the interaction between the human—agent and the building—cyber-physical system (along with its dynamical system). Interestingly, actions of all humans present in the building and the building's dynamical system, potentially influence the actions of the humans who are about to interact in such a cyber-physical system. We assume the humans in the building environment are utility maximizers—agents, either playing according to a Nash strategy [130] or co-optimizing their utility functions (either maximize a complex utility function or play according to a myopic behavior [23, 48, 142, 147]). We model human's utility function by:

i defining a set of base functions leading to a concave utility function and consequently to a unique Nash

ii leveraging classical discrete choice models and implementing a non-linear formulation by adapting efficient machine learning techniques

iii developing a sequential decision-making model using a Deep Learning technique

1.2 Contributions & Challenges

The statement of the dissertation problem can be described as:

This thesis develops efficient learning and optimization schemes for human-centric cyber-physical systems by using spatial and complex information, which arise from building occupants' habits, wellbeing, comfort, productivity, satisfaction, and energy supply variability?

This can briefly be described by key concepts as:

- Allowing building occupants for active—real time integration in a smart-building setting through a gamification approach
- Modeling building occupants' habits and decision-making process using game theoretic approaches
- Develop learning techniques for human-building interactions while improving accuracy, dealing with adversarial attacks (security risks in utility learning procedure), and quantifying robustness in learning
- Leveraging correlations in utility learning for integration into an online utility learning and incentive design scheme
- Develop sequential decision-making models capable to leverage growing amount of data coming from deployed human-centric cyber-physical systems

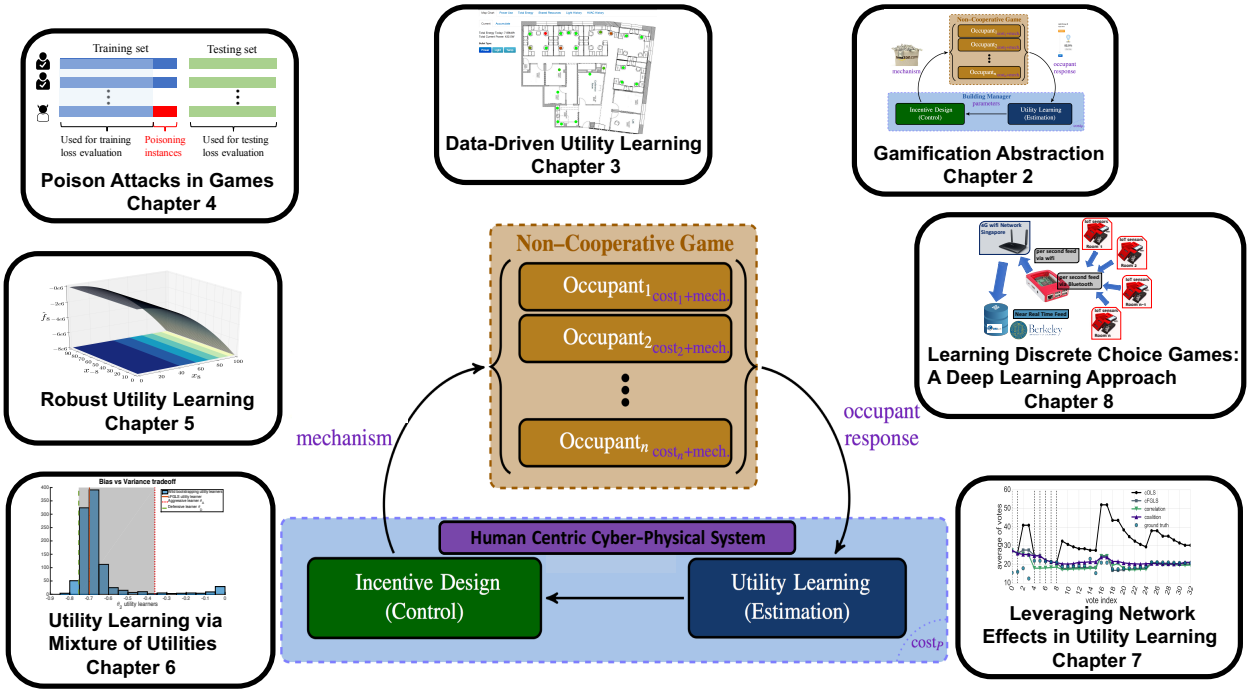


Figure 1.1: Thesis overview: Gamification abstraction in human-centric cyber-physical systems (chapter 2), proposed advanced utility learning frameworks (chapters 3 - 7), and sequential decision-making modeling—utility learning by leveraging Deep Learning (chapter 8).

Figure 1.1 shows thesis overview with highlighted key results and concepts. In the next sections a detailed description is presented.

1.2.1 Gamification for Building Energy Efficiency: Machine Learning & Incentive Design

Human decision-makers play a critical role in the management and operation of contemporary infrastructure, due in no small part to the advent of Internet of Things and cyber-physical sensing/actuation platforms in industry. The cooperation of human agents with building automation in smart infrastructure settings helps improve system's robustness and sustainability by taking advantage of both the control potential for computational methods and the flexibility provided by humans-in-the-loop elements. The adaptability and simultaneous automation inherent to such a system makes it possible to accommodate a variety of dynamic situations that might arise like the automatic shifting of demand during peak energy-usage hours. Put into more broad terms, the goal of many infrastructure systems is to enact system-level efficiency improvements by using a high-level *planner* (e.g. facility manager) to coordinate autonomously acting agents in the system (e.g. selfish human

decision-makers). It is this type of functionality that makes smart building technology so essential to the development of an ideal *smart city*.

We first introduce a novel gamification abstraction [149] towards smart-building energy efficiency. A social game is proposed for encouraging energy efficient behavior in occupants by distributing points which determine the likelihood of winning in a lottery. On top of the designed game we propose a closed-loop human-centric cyber-physical system by estimating occupants' utilities and formulating the interaction between the building manager and the occupants as a reversed Stackelberg game—hierarchical control architecture in which there are multiple followers that play in a non-cooperative game. In such a setting the estimated utilities are used for determining the occupant behavior in the non-cooperative game and concurrently to design optimal incentives for boosting system's efficiency. Due to nonconvexities and complexity of the problem, in particular the size of the joint distribution across the states of the occupants, we propose a solution to this bi-level optimization problem by using a particle swarm optimization method.

Our bi-level approach to efficient building energy management focuses on residential—commercial buildings and utilizes cutting-edge Internet of Things (IoT) sensors and cyber-physical systems sensing/actuation platforms. We design social games aimed at incentivizing occupants to modify their behavior so that the overall energy consumption is reduced. In certain living situations, occupants in residential—commercial buildings are not responsible for paying for the energy resources they consume. Hence, there is often an imbalance between the incentives of the facility manager and the occupants. The competitive aspects inherent to the social game framework motivate occupants to address their inefficiencies as well as encourage responsible energy usage on an individual basis.

At the core of our approach is the decision to model the occupants as non-cooperative agents who play according a Nash strategy [130], a sequential discrete choice game [173] or co-optimize their utility functions (either maximize a complex utility function or playing according to a myopic behavior [23, 48, 142, 147]). Specifically, we model decision-making agents as utility maximizers. Agents are strategic entities that make decisions based on their own preferences without consideration of others. The game-theoretic framework both allows for qualitative insights to be made about the outcome of aforementioned selfish behavior—more so than a simple prescriptive model—and, more importantly, can be leveraged in designing mechanisms for incentivizing agents.

The broader purpose of this chapter is to introduce a general framework that leverages game-theoretic concepts to learn models of players' decision-making in residential—commercial buildings provided with our implementation of a novel energy social game. The game theoretic framework both allows for qualitative insights to be made about the outcome of such selfish behavior and, more importantly, can be leveraged in designing *incentives* for adjusting agents strategies.

1.2.2 Data-Driven Utility Learning in Games: A Nash Approach

After introducing a general framework for gamification using a variety of game theoretic models, we formulate the core utility estimation problem wrapped in a convex optimization formulation by using first and second order necessary conditions for Nash equilibria. In particular, for the main estimation setting the gradient of each human's utility function should be identically zero at the observed Nash equilibrium. This is the case since the observed Nash equilibria are all inside the feasible region so that none of the constraints are active, i.e. we do not have to check the derivative of Lagrangian of each occupant's optimization problem. Particularly, the second order conditions help the restriction of parameter space of defined utility functions as to ensure concavity—unique Nash equilibrium under an N-person concave game [152].

For each observation, we assume that it corresponds to occupants playing a strategy that is approximately a Nash equilibrium and also is considered a data point for the overall utility learning framework. The result is as a constrained Ordinary Least Squares (cOLS) problem with classical spherical noise error terms—Gaussian distributed residual residuals (homoscedasticity). Since it is a constrained optimization problem (Box-constrained least squares), a closed form solution cannot be attained but an iterative solution with projections to parameter space [29, 52] is used. Despite the given assumptions under this core data-driven utility learning, it serves well as a basis for robust (chapter 5) and more advanced hierarchical models (chapter 6).

1.2.3 Poisoning Attacks on Utility Learning in Games

As one of the central questions in game theory deals with predicting the behavior of an agent (chapter 3), this question has led to the development of a variety of theories and concepts, such as Nash equilibrium, for predicting agents' actions according to their utilities. These predictions may, in turn, be used to inform economic and engineering decisions. Often in game-theoretic analysis, it is assumed that there exists at the best a well-known model for each individual's agent's utility. However, the model primitives are not directly observable and can be difficult to estimate (chapters 3 - 4). Small errors in the estimates may substantially affect the predictions of agents' behaviors. Thus, it is crucial to develop accurate estimates of utility models.

Along with increased reliance on data come greater security risks. Recent research in machine learning community has shown the vulnerability of data-driven algorithms in various applications [39, 53, 112, 117]. Adding a few well-crafted data points into the training dataset (poisoning attack) or manipulating the testing data (evasion attack) can induce spurious model outputs. These datasets are often crowdsourced or collected from wireless sensor networks, and therefore easy to falsify. High-profile attacks on spam filtering [113], recommendation system [114], face recognition [17] among others have diminished the credibility of data-driven technologies. In parallel to the study of attack and defensive strategies in the

machine learning community, this paper presents the first effort to investigate the effectiveness of data-driven models used for game-theoretical analysis under adversarial conditions.

While game theory in a general gamification abstraction (chapter 2) has been employed for modeling humans—agents' decisions, game theoretic analysis often assumes that the utility function of each agent is known *a priori*, and yet this assumption does not hold for many real-world applications. The combination of Internet of Things (IoT) and advanced data analysis techniques has stimulated fruitful research on learning agents' utility functions from data. Just as many other data-driven methods, utility learning also suffers from potential security risks [69]. Due to the great economic value of accurate forecasting of agents' behaviors, there are huge incentives for adversaries to attack utility learning methods by poisoning training datasets and mislead predictions to achieve malicious goals. Under a brute-force (no robust) learning scheme (chapter 3) we introduce and analyze optimal poisoning attack strategies in order to understand adversarial actions and further encourage potential defenses. Moreover, we study how an adversary might disguise the attacks by mimicking normal actions. We apply the proposed privacy preserving optimal strategy to compute action for users with synthetic data and show that with small perturbation added, users can protect their privacy as well as keep high utility. We also conduct experiments for real-world social energy game and show that the proposed optimal strategy can make nice tradeoff between privacy and user utility. More interestingly, we develop an algorithm to synthesize malicious attack points that imitate normal behaviors and are thereby hard for a defender to detect.

1.2.4 Robust Utility Learning Framework via Inverse Optimization

Modeling human interaction as a continuous game between non–cooperative players (chapter 2), we develop a robust parametric utility learning framework that employs constrained feasible generalized least squares estimation with heteroskedastic inference [99]. The resulting estimators are consistent with minimized variance and bias in which estimator variance is reduce. To improve forecasting performance, we extend the robust utility learning scheme by employing wild parametric bootstrapping with ensemble learning methods like: bagging, bumping—a stochastic search technique, and gradient boosting. The developed framework shows significant improvement over in our estimation accuracy especially under Constrained Ordinary Least Squares (cOLS) estimations (chapter 3).

We assume a parametric form of utility function for each player that is dependent on the decisions of others. Correlations between players' decisions are not known *a priori*. Assuming observations are approximately Nash equilibria, we use first– and second–order conditions on player utility functions to construct a constrained regression model. The result is as a constrained Generalized Least Squares (cGLS) problem with non-spherical noise error terms. Using constrained Feasible Generalized Least Squares (cFGLS), an implementable version of cGLS, we utilize heteroscedasticity inference to approximate the correlated errors.

Noting that data sets of observed decisions often may be small relative to the number of model parameters in practice, we employ bootstrapping to generate pseudo-data from which we learn additional estimators. The bootstrapping process allows us to derive an asymptotic approximation of the bias and standard error of an estimator. We utilize ensemble methods such as bagging, bumping, and gradient boosting to extract an estimator from the pseudo-data generated estimators that results in a reduced forecasting error. The ensemble methods are robust under noise and autocorrelated error terms. We apply the robust utility learning framework to a model of Bertrand-Nash competition between firms in order to illustrate the framework and its performance. Also, we test our framework using data resulting from a real-world social energy game and show that the proposed robust learning method achieves excellent forecasting accuracy.

1.2.5 Utility Learning via Mixture of Probabilistic Hierarchical Utilities

In several previous developed tools [103, 104, 149] for estimating parameters of users' utility functions and designing incentives to encourage socially optimal and efficient behaviors, the core approach is the fact that the agents are modeled as non-cooperative agents who play according to a *Nash equilibrium strategy*. This serves the purpose of modeling the agents as strategic entities who make decisions based on their own preferences in spite of others. In [102], we extended the basic utility learning framework to a robust framework. In particular, assuming a parametric form of utility function for each agent, we utilized constrained Feasible Generalized Least Squares (cFGLS) to formulate a parameter estimation scheme in which the estimator variance is reduced, unbiased, and consistent.

We develop an adaptation of *mixture of regression models* that takes in to account heteroskedasticity and we show the performance of the proposed method is more accurate than robust utility learning methods such as constrained Feasible Generalized Least Squares (cFGLS) (chapter 5), ensemble methods such as bagging (chapter 5), and classical methods such as Ordinary Least Squares (OLS) (chapter 3). Not surprisingly a novel framework for estimating a mixture of utility functions is developed by extending the classical *mixture of regression models* framework to inverse modeling of utility functions in non–cooperative, continuous games.

We introduce a framework of dynamic decision-making by evaluating softmax functions over non-cooperative agents' past actions while assigning utility functions with time-varying parameters. In this new method of parametric utility learning for non-cooperative [100], continuous games using a probabilistic interpretation for combining multiple utility functions—thereby creating a *mixture of utilities*—under non-spherical noise terms is introduced. A generalized model of *mixture of utilities* can be defined leveraging *Hierarchical Mixture of Experts Models* [87]. Under this model, softmax gates assign "soft" thresholds and give a hierarchical probabilistic interpretation at the decision process.

In particular, we present the theoretical formulation of a new parametric utility learning

method that uses a probabilistic interpretation—i.e. a mixture of utilities—of agent utility functions. The mixture of utilities modeling paradigm allows us to account for variations in agents' parameters over time. Our method combines resulting utility learners under non-spherical noise terms. The main contribution is the adaptation to utility learning model of a probabilistic framework using a softmax function, which used for combining regression models. The resulting scheme is a *Mixture of constrained Feasible Generalized Least Squares* (Mix-cFGLS) which uses heteroskedasticity inference for correlated errors in the resulting regression model. Mix-cFGLS is a statistical model that we show is a powerful tool for utility learning that providing greater accuracy in the prediction of players' responses. Furthermore, captured in this framework is the fact that players' utility functions are not static; instead, the parameters of players utility functions can depend on historical data. In the Mix-cFGLS framework, we explore the tradeoff between minimizing bias and minimizing the variance of predictions. We show that by using Mix-cFGLS for utility learning, allowing for a small amount of bias results in a substantial decrease in variance and increase in forecasting accuracy. Moreover, the benefit of using a mixture of utilities framework is that it allows us to capture the effects of different environmental conditions on the outcome of decisions and alongside temporal dependencies in the data process. For example, perhaps one utility function better models a person in the morning and another in the evening given concurrently other persons utility functions—action set; through mixing multiple utilities, we can capture this behavior in a single model. Bias assignment to utility learners occurs using a probabilistic interpretation by evaluating softmax functions given non-cooperative agents past actions.

1.2.6 Leveraging Network Effects in Utility Learning

Wanting to reduce the complexity of existing methods while increasing the forecasting accuracy by accounting for possible collusions between decision makers, we design a new method of data-driven modeling of human decision-making. Building on existing game theoretic concepts such as *coalition games* [133], we are able to extend our existing robust utility learning framework [99, 102] to a utility learning framework that has the potential to leverage interactions amongst players in order to reduce complexity (amortize time has a similar complexity with constrained ordinary least squares solutions—chapter 3) and even operate online—thereby allowing for adaptive incentive mechanisms to be implemented[1].

In particular, we propose a framework to estimate correlations between players and propose two utility learning methods that leverage these correlations. The first method is the *correlation utility learning framework* in which we use estimated correlations to define a *correlation game* in which each player's utility function is converted into a *correlation utility*. This method is described in two steps: in the first step, we apply constrained Feasible Generalized Least Squares (cFGLS) with noise estimation (which is the core of our robust utility

[1]Even if we do not explore online incentive design the work we present is the basis for further research in that direction.

learning method [99, 102]) to estimate the correlations between players. This first step is deployed in a small part of the given dataset, which ensures reduced overall complexity (in amortize setting). In the second step, we construct correlation utilities by taking a weighted sum of each player's constrained Ordinary Least Squares (cOLS) estimated utility function and all other players' estimated utilities that are highly correlated with it. We optimize over the weights to improve the forecast of players' decisions.

We refer to the second utility learning method as the *coalition utility learning framework*. This method can also be described in two steps: in the first step, a small subset of data is used to estimate correlations between players—again using cFGLS with noise estimation—that are then used to define coalitions amongst the most positively correlated players. In the second step, we employ a simple cOLS estimation procedure to estimate the parameters of the coalition utilities. The proposed method, being based on cOLS, is amenable to conversion to an online estimation scheme which can be integrated into an adaptive incentive design framework [147]. In addition to the estimated correlations between players resulting from cFGLS with noise estimation we deploy a *Shapley value regression feature selection scheme* in the dataset resulting from gaming data process. Interestingly, Shapley value technique gives a clear view of agents coalitions and generates a generalized method for building such games.

1.2.7 Utility Learning Under Discrete Choice Games: A Deep Learning Approach

In previous chapters, an arsenal of advanced learning techniques are presented for non-cooperative continuous games assuming humans play according to a Nash strategy. On the other hand, we are motivated to enable humans in a gamification setting in which they don't immediately compete with others but they co-optimize their own utility functions across a set of discrete actions. We propose the design and implementation of a large-scale network game with the goal of improving the energy efficiency of a building through the utilization of cutting-edge Internet of Things (IoT) sensors and cyber-physical systems sensing/actuation platforms [98]. By observing human decision-makers and their decision strategies in their operation of building systems, we can apply inverse learning techniques in order to estimate their utility functions. As in most game theoretic analysis the utility function of each agent seem to be known a priori, we propose a benchmark utility learning framework that employs robust estimations for classical discrete choice models provided with high dimensional imbalanced data emerging from many real-world applications. To improve forecasting performance and dealing with high-dimensional datasets, we extend the benchmark utility learning scheme by leveraging Deep Learning end-to-end training with Deep bi-directional Recurrent Neural Networks.

At the core of our approach is the decision to model the occupants as non-cooperative agents who play according to a sequential discrete choice game. Discrete choice models have been used extensively to examine modes of transportation [172], choice of airport [10], de-

mand for organic foods [55], robbery patterns [11], and even school social interactions [165]. Under this assumption of non-cooperation, we were able to use a randomized utility framework and propose novel utility estimation algorithms for the occupants' utility functions. Most importantly, estimating agent utility functions via our method results in predictive models that provide excellent forecasting of occupant actions—usage.

Our framework is centered around learning agent preferences over room resources such as lighting as well as external parameters like weather conditions, high-level grid control, and provided incentives. Specifically, we model decision-making agents as sequential utility maximizers. Agents are strategic entities that make decisions based on their own preferences without consideration of others. The game-theoretic framework both allows for qualitative insights to be made about the outcome of aforementioned selfish behavior—more so than a simple prescriptive model—and, more importantly, can be leveraged in designing mechanisms for incentivizing agents.

Moreover, we provide a demo web portal for demonstrating our infrastructure and for downloading de-identified high dimensional data sets[2]. High-dimensional data sets can serve either as a benchmark for discrete choice model learning schemes or as a great source for analyzing occupants' usage in residential buildings. Towards this scope, we use conventional deep variational auto-encoders [94] or recurrent network based adaptation of variational auto-encoders [57] as an approach to create a nonlinear manifold (encoder) that can be used as a generative model. Variational auto-encoders can fit large high dimensional data sets (like our social game application) and train a deep model to generate data like the original data set. In a sense, generative models automate the natural features of a data set and then provide a scalable way to reproduce known data. This capability can be employed either in the utility learning framework for boosting estimation or as a general way to create simulations mimicking occupant behavior—preferences.

1.3 Outline

Specifically, in Chapter 2, we introduce a generalized gamification framework for towards energy efficiency with core learning schemes & incentive design approaches. Following in Chapter 3 is given the base utility learning optimization formulation, which seems vulnerable in attacks as it is given in Chapter 4. In contrast to Chapter 3, a robust parametric estimation framework for continuous games is given in Chapter 5 along with ensemble learning adaptations for improving overall forecasting accuracy. Hierarchical probabilistic models in Chapter 6 are generalize our utility learning framework with extraordinary accuracy improvements while leveraging correlations among agents in the learning scheme gives a great tradeoff between computation complexity at the estimation phase and forecasting accuracy (chapter 7). Finally, in Chapter 8, we develop a sequential decision-making model utilizing Deep Learning techniques leveraging a growing amount of data in a novel proposed human-centric cyber-physical system.

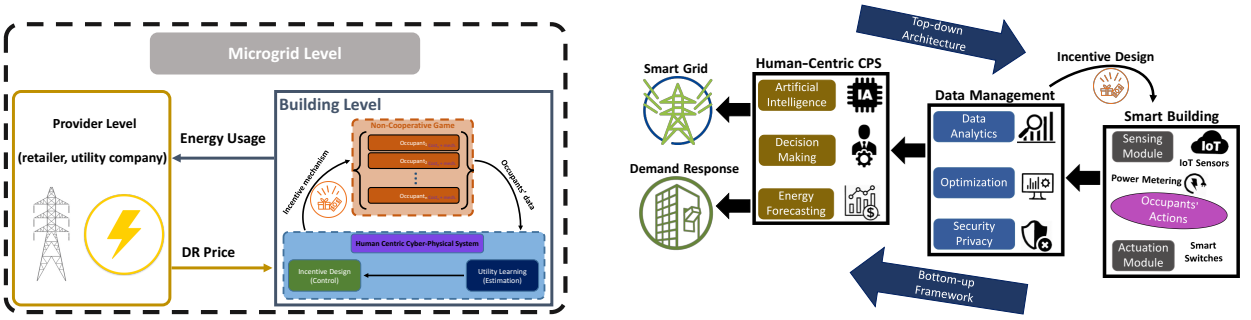
[2] *smartNTU* demo web-portal: https://smartntu.eecs.berkeley.edu

Chapter 2

Gamification for Building Energy Efficiency: Machine Learning & Incentive Design

This chapter provides a generalized gamification abstraction towards smart-building energy efficiency (figure 2.1) and connection to demand response pricing in smart grid applications. Building occupants' and manager's interaction is modeled as a reversed Stackelberg game (leader—followers) in which there are multiple followers (building occupants) that play in a non-cooperative game. Leader's overall task is to coordinate followers. This overall novel design is a closed-loop human-centric cyber-physical system that leverages advanced machine learning and Deep Learning algorithms for utility estimation regarding the determination of individual occupant—follower behavior in the non-cooperative game. The proposed abstraction takes in account building manager's (leader) cost efficient function and designs appropriate incentives towards more desirable outcomes. Overall, the derived gamification abstraction simultaneously learns the decision-making process of agents via utility learning schemes (using statistical learning theory) that we developed and adaptively designing economic mechanisms to both elicit informative responses from agents and incentivize desirable outcomes. The framework supports learning agents' preferences over shared resources as well as understanding how preferences change as a function of external stimuli such as physical control or incentives. Such a framework can be used in the design of incentive mechanisms that realign agents' preferences with those of the planner—which often represent system-level performance criteria—through fair compensation.

Section 2.1 introduces the idea of gamification and provides related work-applications in both academic—industrial setting. Sections 2.2 and 2.3 present a unified framework capable of enabling users' engagement, modeling their interaction and being used to define machine learning algorithms for updating users' utility estimation. Section 2.2 covers several game theoretic frameworks like Nash-play, and sequential discrete choice games. Lastly, Section 2.3 formulates a bi-level optimization problem for solving optimizing leader's cost function while updating the estimates of followers decision model.



(a) Gamification abstraction for *Human-Centric Cyber-Physical Systems*

(b) Overview of block diagram—research design framework

Figure 2.1: Block diagram & gamification abstraction for human-building interaction

2.1 Gamification & its Applications

Smart grid technology is focused on enabling efficient grid integration and comprehensive analysis to support advances in renewable energy sources, power systems management, minimization of inefficiencies in usage, and maximization of user savings. However, challenges in power grid applications, such as the lack of predictability, and the stochastic environment in which the power grid operates complicate the synthesis of an all-encompassing solution. To address these problems, industry and researchers in the fields of power grid design and control have put forth considerable research and development efforts in smart grid design, demand-side management, and power system reliability.

With the help of digital hardware and information technology, smart grid design relies more and more on the development of decision-capable intelligence in the context of grid automation. Novel methods [156] for smart grid design incorporate real-time analysis and stochastic optimization to provide power grid observability, controllability, security, and an overall reduction of operational costs. Specifically, the integration of data analytics and innovative software platforms have led to effective trends in demand-side management [141] and demand response [120]. These studies explored and drew upon methods from behavioral decision research to analyze the decision patterns of users and consumers. The simulations and empirical results from these studies reinforce the significance of forecasting energy demands and the potential advantages of managing these demands by leveraging models of intelligent decision-makers. In this context, we can see that the process of modeling and predicting the actions of decision-makers in the control of large networks is a significant development toward improving the operational efficiency of smart grids.

Game theory can serve as an extremely useful tool for the real-time forecasting of decision-makers in an interactive setting. Classical models from game theory allow for qualitative insights about the outcome of scenarios involving the selfish behavior of competitive agents and can be leveraged in the design of incentives for influencing the goals of these agents. Contemporary research in the energy and power systems domain leverages game theoretic

models in a multitude of applications. As previously mentioned, these types of approaches have been implemented in the modeling of various aspects of smart grid control. Specifically, we can observe instances of game theory applications in the context of smart grid demand response programs using methods such as supply-balancing [188], hierarchical Stackelberg game settings [187], and Vickrey-Clarke-Groves (VCG) auction mechanisms [159].

The use of game theoretic models creates new avenues for modeling dynamic economic interactions between utility providers and consumers inside a distributed electricity market [189]. Another example study is the investigation of crowdfunding as an incentive design methodology for the construction of electric vehicle charging piles [193]. Game theory has also been directed toward the optimal design of curtailment schemes that control the fair allocation of curtailment among distributed generators [3]. Expanding on previous work, researchers have studied game theory applications in the context of incentive-based demand response programs for customer energy reduction as well [178]. In these types of applications, customer interaction with the incentive provider is modeled using game theoretic models while their engagement is represented probabilistically.

In the majority of the previously discussed game theoretic modeling applications, results are generated purely by simulation without the use of real data. Furthermore, previous applications neglect to propose any novel techniques for learning the underlying utility functions that dynamically predict strategic actions. Due to these limitations, we cannot reasonably expect to learn (or estimate) user functions in a gaming setting nor generalize results to broader scenarios. In real-life applications, the game theoretic models are not known a priori; therefore, the developed methods should have some way to account for data-driven learning techniques. In our past work, we have explored utility learning and incentive design as a coupled problem both in theory [99, 69] and in practice [149] under a Nash equilibrium model. Our utility learning approaches are presented in a platform-based design flow for smart buildings [67]. The broader purpose of this paper is to present a general learning framework that leverages game theoretic concepts for learning models of building occupant decision making in a competitive setting and under a discrete set of actions.

In Figure 2.1, we present a block diagram of our proposed research design framework toward building energy management from both a top-down and bottom-up perspective, motivated by previous model illustrations [67]. The block diagram consists of three layers: the smart building layer, the data management layer, and the top human-centric cyber physical systems layer. Each layer has some connection between its respective components and upper level abstractions. From the proposed bottom-up framework, the aggregated occupant patterns are processed and passed to an artificial intelligence layer that is capable of real-time energy forecasting, which can then be integrated with applications like demand response programs. Through optimization and data analysis, the proposed design framework leverages advanced incentive design schemes aimed at engaging smart building occupants. In addition, the data management layer provides the opportunity to implement security and privacy protocols against malicious attacks [69].

Contemporary building energy management techniques employ a variety of algorithms in order to improve performance and sustainability. Many of these approaches leverage ideas

from topics such as optimization theory and machine learning. Our goal was to improve building energy efficiency by introducing a gamification system that engages users in the process of energy management and integrates seamlessly through the use of a human-centric cyber-physical framework. There exists a considerable amount of previous work demonstrating the success of control and automation in the improvement of building energy efficiency [6, 20, 118]. Some other notable techniques implement concepts such as incentive design and adaptive pricing [122, 151]. Modern control theory has been a critical source of inspiration for several approaches that employ ideas like model predictive and distributed control and have demonstrated encouraging results in applications like HVAC. Unfortunately, these control approaches lack the ability to consider the individual preferences of occupants, which highlights a significant advantage of human-centric scenarios over contemporary methods. This trend is also apparent in machine learning approaches to building utility management. While these machine learning approaches are capable of generating optimal control designs, they fail to adjust to occupant preferences and the associated implications of these preferences to the control of building systems. The heterogeneity of user preferences in regard to building utilities is considerable and necessitates a system that can adequately account for differences from occupant to occupant. In general, the presence of occupants greatly complicates the determination of an efficient building management system. With this in mind, focus has shifted toward modeling occupant behavior within the system in an effort to incorporate their preferences. To accomplish this task, the building and its occupants are represented as a multi-agent system targeting occupant comfort [20]. First, occupants and managers are allowed to express their building preferences, and these preferences are used to generate an initial control policy. An iteration on this control policy is created by using a rule engine that attempts to find compromises between preferences. Some drawbacks of this control design are immediately apparent. There should be some form of communication to the manager about occupant preferences. In addition, there exists no incentive for submission of true user preferences, and no system is in place for feedback from occupants.

Gamification [37] serves well for boosting user overall engagement and targeting to optimal behaviors. Our key approach is the implementation of a gamification abstraction (figure 2.1), which is deployed in the form of a **social game** among users in a non-cooperative setting. Similar methods that employ *social games* have been applied to transportation systems with the goal of improving flow [127, 135]. Another example application can be found in the medical industry in the context of privacy concerns versus expending calories [15]. Entrepreneurial ventures have also sought to implement solutions of their own to the problem of building energy efficiency. *Comfy*[1], *Cool Choices*[2] and *Keewi*[3] are examples of start-ups that have developed interesting approaches to controlling building utilities. *Comfy*'s product is primarily data-driven, while allowing occupants to have flexibility and independence in their control of HVAC. *Keewi* has developed a product for plug load energy management

[1]https://comfyapp.com
[2]https://coolchoices.com/how-it-works/improve
[3]https://www.keewi-inc.com/index.php

and real-time energy monitoring using gamification approaches. On the other hand, *Cool Choices* has designed a game theoretic framework centered around improving sustainability. Representing the larger industry agents, companies such as *Oracle - OPower*[4] are attempting to balance energy efficiency and demand response management in an effort to engage customers digitally. Finally, it has been shown that societal network games are useful in a *smart city* context for improving energy efficiency and human awareness [31, 36].

The idea behind gamification—social game context is to create friendly competition between occupants. In turn, this competition will help motivate them to individually consider their own energy usage and, hopefully, seek to improve it. This same technique of gamification has been used as a way to educate the public about energy usage [8, 96, 132, 157]. It has also been cleverly implemented in a system that presents feedback about overall energy consumption to occupants [162]. One case of a gamification methodology was used to engage individuals in demand response (DR) schemes [115].

Each of the users is represented as a utility maximizer within the model of a Nash equilibrium where occupants gain incentive for reduction in consumption during DR events. In contrast to approaches that target user devices with known usage patterns [115], our approach focuses on a general abstraction which can incorporate shared sources (Nash approach), and (or) personal utilities—sources such as lighting without initial usage information, simulating scenarios of complete ignorance to occupant behaviors. For our method, we leverage past user observations to learn the utility functions of individual occupants by the way of several novel algorithms. Through this approach, we can generate excellent prediction of expected occupant actions. Our unique social game methodology simultaneously learns occupant preferences while also opening avenues for feedback. This feedback is translated through individual surveys that provide opportunities to influence occupant behavior with adaptive incentive. With this technique, we are capable of accommodating occupant behavior in the automation of building energy usage by learning occupant preferences and applying a variety of novel algorithms. Furthermore, the learned preferences can be adjusted through incentive mechanisms to enact improved energy usage.

A series of experimental trials were conducted to generate real-world data, which was then used as the main source of data for our approach. This differentiates our work from a large portion of other works in the same field that use simulations in lieu of experimental methods. In many cases, participants exhibit a tendency to revert to previously inefficient behavior after the conclusion of a program. Our approach combats this effect by implementing incentive design that can adapt to the behavior and preferences of occupants progressively, which ensures that participants are continuously engaged. From a managerial perspective, the goal is to minimize energy consumption while maximizing occupant comfort. With this social game framework, the manager is capable of considering the individual preferences of the occupants within the scope of the building's energy consumption. The advent of this social game system could potentially offer an unprecedented amount of control for managers without sacrificing occupant comfort and independence.

[4]https://opower.com

2.2 Generalized Follower Game

In our core abstraction we model the interaction between the building manager (leader) and the occupants (followers) as a leader-follower type game. We use the terms leader and building manager interchangeably and, similarly, for follower and occupant.

In this model the followers are utility maximizers that play in a non-cooperative game for which we use several game theoretic models. Followers can play according to the Nash equilirbium concept [130], a sequential discrete choice game [173] or co-optimize their utility functions (either maximize a complex utility function or playing according to a myopic behavior [23, 48, 142, 147]). In our abstraction the leader is also an utility maximizer with an utility that is dependent on the choices of the followers. The leader can influence the equilibrium of the game amongst the followers through the use of incentives which impact the utility and thereby the decisions of each follower. For instance, provided that the leader desires to reduce the energy consumption in the building, then can formulate a model of how the occupants make decisions about their energy usage.

We begin by describing the game-theoretic framework used for modeling the interaction between the occupants. We remark that the use of game theory for modeling the behavior of the occupants has several advantages. First, it is a natural way to model agents competiting over scarce resources. It can also be leveraged in the design of incentives for behavioral change in that it incorporates the ability to model the occupants as strategic players.

2.2.1 Follower Game: A Nash Approach for Agent's Decision-Making Model

In this section, we abstract the agents' decision-making processes in a game–theoretic framework assuming that agents play according to a Nash strategy.

Consider p agents[5]—i.e. decision-making entities—indexed by the set $\mathcal{I} = \{1, \dots, p\}$. Each agent is modeled as a *utility maximizer* that seeks to select $x_i \in \mathbb{R}$ by optimizing

$$f_i(x_i, x_{-i}) = f_i^{\text{nom}}(x_i, x_{-i}) + f_i^{\text{inc}}(x_i, x_{-i}). \tag{2.1}$$

where $f_i^{\text{nom}}(x_i, x_{-i})$ and $f_i^{\text{inc}}(x_i, x_{-i})$ are the nominal and incentive components, respectively, of agent i's utility function and where $x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \in \mathbb{R}^{n-1}$ is the collective choices of all agents excluding the i–th agent[6].

The choice x_i abstracts the agent's decision; it could represent, e.g., how much of a particular resource they choose to consume. The nominal component of f_i captures the agent's individual preferences over x_i and may depend on the decisions of others x_{-i}. The incentive component models the portion of the agent's utility that can be designed by the planner; it also may depend on the decisions of other agents.

[5] We refer to the decision-makers as *agents* and use the term interchangeably with *players*.

[6] Note that while for notational simplicity we assume that $x_i \in \mathbb{R}$, the work easily extends to a higher dimensional choice vector for each agent.

Agent i's optimization problem is also subject to constraints; the constraint set is given by $\mathcal{C}_i = \{x_i |\ h_{i,j}(x_i) \geq 0, j = 1, \ldots, \ell_i\}$ where each $h_{i,j}$ is assumed to be a concave function of x_i. Such constraints may encode cyber or physical constraints arising from the underlying system—in the social game example presented in Section 4, we will see that these constraints are physical bounds. Thus, given x_{-i}, agent i faces the following optimization problem:

$$\max\{f_i(x_i, x_{-i}) |\ x_i \in \mathcal{C}_i\}. \tag{2.2}$$

The game $(f_1, \ldots, f_p)$ is a continuous game on a convex strategy space $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_p$. To model the outcome of the strategic interactions of agents, we use the *Nash equilibrium* concept.

Definition 1. *A point $x \in \mathcal{C}$ is a Nash equilibrium for the game $(f_1, \ldots, f_p)$ on $\mathcal{C}$ if, for each $i \in \mathcal{I}$,*

$$f_i(x_i, x_{-i}) \geq f_i(x_i', x_{-i}) \ \ \forall\ x_i' \in \mathcal{C}_i. \tag{2.3}$$

We say $x \in \mathcal{C}$ is an ε–Nash equilibrium for $\varepsilon > 0$ if the above inequality is relaxed:

$$f_i(x_i, x_{-i}) + \varepsilon \geq f_i(x_i', x_{-i}) \ \ \forall\ x_i' \in \mathcal{C}_i. \tag{2.4}$$

We say a point is a *local Nash equilibrium* (respectively, a *ε–local Nash equilibrium*) if there exists $W_i \subset \mathcal{C}_i$ such that $x_i \in W_i$ and the above inequalities hold for all $x_i' \in W_i$.

If each f_i is concave in x_i and $\mathcal{C}$ is convex, then the game is a p–person concave game. In the seminal work by Rosen [152], it was shown that a (pure) Nash equilibrium exists for every concave game.

Theorem 1 (Rosen [152])**.** *A Nash equilibrium exists for every concave n-person game.*

Define the Lagrangian of each player's optimization problem as follows:

$$L_i(x_i, x_{-i}, \mu_i) = f_i(x_i, x_{-i}) + \sum_{j \in A_i(x_i)} \mu_{i,j} h_{i,j}(x_i) \tag{2.5}$$

where $A_i(x_i)$ is the active constraint set at x_i. We can define

$$\omega(x, \mu) = \begin{bmatrix} D_1 L_1(x, \mu_1) \\ \vdots \\ D_n L_n(x, \mu_n) \end{bmatrix} \tag{2.6}$$

where $D_i L_i$ denotes the derivative of L_i with respect to x_i.

The Lagrangian of agent i's optimization problem is given by

$$L_i(x_i, x_{-i}, \mu_i) = f_i(x_i, x_{-i}) + \textstyle\sum_{j \in \mathcal{A}_i(x_i)} \mu_{i,j} h_{i,j}(x_i) \tag{2.7}$$

where $\mathcal{A}_i(x_i)$ is the active constraint set at x_i and $\mu = (\mu_1, \ldots, \mu_p)$ with $\mu_i = (\mu_{i,j})_{j=1}^{\ell_i}$ are the Lagrange multipliers. Assuming appropriate smoothness conditions on each f_i and $h_{i,j}$,

the differential game form [144],[147]—which characterizes the first–order conditions of the game—is given by

$$\omega(x,\mu) = [D_1 L_1(x,\mu_1)^\top \ \cdots \ D_p L_p(x,\mu_p)^\top]^\top \tag{2.8}$$

where $D_i L_i$ denotes the derivative of L_i with respect to x_i.

Consider agent i's optimization problem (2.2) with x_{-i} fixed and where each f_i and $h_{i,j}$ for $j \in \{1,\dots,\ell_i\}$, $i \in \mathcal{I}$ are concave, twice continuously differentiable functions. Then, assuming an appropriate constraint qualification condition [13], the necessary and sufficient conditions for optimality of a point x_i are as follows: there exists $\mu_i \in \mathbb{R}_+^{\ell_i}$ such that:

(i) $D_i L_i(x,\mu_i) = 0$; (ii) $\mu_i h_{i,j}(x_i) = 0$ for each $j \in \{1,\dots,\ell_i\}$; (iii) $h_{i,j}(x_i) \geq 0$ for each $j \in \{1,\dots,\ell_i\}$.

Regardless of the concavity assumption, the point x_i is a local maximizer if $\mu_{i,j} > 0$ and $z^\top D_{ii}^2 L_i(x,\mu_i) z < 0$ for all $z \neq 0$ such that $D_i h_{i,j}(x_i)^\top z = 0$ for $j \in A_i(x_i)$. Such conditions motivate the following definition.

Definition 2 (Differential Nash Equilibrium)**.** *Consider a game $(f_1,\dots,f_p)$ on $\mathcal{C}$ where f_i and $h_{i,j}$ for each $j \in \{1,\dots,\ell_i\}$ and $i \in \mathcal{I}$ are twice continuously differentiable. A point $x \in \mathcal{C} \subset \mathbb{R}^p$ is a differential Nash equilibrium if there is a $\mu \in \mathbb{R}^{\sum_{i=1}^p \ell_i}$ such that the pair (x,μ) satisfies:*

(i) $\omega(x,\mu) = 0$; (ii) for each $i \in \mathcal{I}$, $z^\top D_{ii} L_i(x,\mu_i) z < 0$ for all $z \neq 0$ such that $D_i h_{i,j}(x_i)^\top z = 0$, and $\mu_{i,j} > 0$ for $j \in A_i(x_i)$. If, for a given $\varepsilon > 0$, $\omega(x,\mu) = \varepsilon$ with all the other conditions being satisfied, then x is a ε–differential Nash equilibrium.

The above definition extends the definition of a differential Nash (if we restrict to Euclidean spaces), first appearing in [144], to constrained games on Euclidean spaces. Using this definition, we can also extend Proposition 1 of [144], again where strategy spaces are restricted to be subsets Euclidean.

Proposition 1. *A differential Nash equilibrium of the p–person concave game $(f_1,\dots,f_p)$ on $\mathcal{C}$ is a local Nash equilibrium.*

Proof of Proposition 1. Suppose the assumptions hold. The constraints for each player do not depend on other players' choice variables. We can hold x_{-i}^* fixed and apply Proposition 3.3.2 [13] to the i-th player's optimization problem $\max\left\{f_i(x_i, x_{-i}^*) \mid x_i \in \mathcal{C}_i\right\}$. Since each f_i is concave and each $\mathcal{C}_i$ is a convex set, x_i^* is a global optimum of the i-th player's optimization problem under the assumptions. Since this is true for each of the $i \in \{1,\dots,n\}$ players, x^* is a Nash equilibrium. □

The proposition says that the conditions of Definition 2 are sufficient for a local Nash. In contrast to single-agent optimization problems, for games, the second order conditions do

not imply the equilibrium is isolated [144]. A sufficient condition guaranteeing that a Nash equilibrium x is isolated is that the Jacobian of $\omega(x, \mu)$, denoted $D\omega(x, \mu)$, is invertible [147].

We use (necessary and sufficient) optimality conditions on individual player optimization problems holding other players' strategies fixed to formulate the utility learning framework.

2.2.1.1 Nash Equilibrium Computation: A Dynamical Systems Perspective

We can take a dynamical systems perspective in order to come up with a method for computation of the Nash equilibrium (see, e.g. [43], [142], [152]). We first write down a reasonable set of dynamics, then we show that a Nash equilibrium is a stable fixed point of these dynamics, and finally we suggest a sub-gradient projection method for computation.

It is natural to consider computing Nash equilibria by following the gradient of each occupant's utility function. Hence, we consider the dynamical system obtained by taking the derivative with respect to their choice variable of the Largrangian's for each occupant's optimization problem.

Due to the fact that our constraint set is convex, closed and bounded in $\mathbb{R}^n$ and there is a point in its strict interior, we satisfy a constraint qualification condition which is a sufficient condition for the Karush-Khun-Tucker (KKT) conditions for each occupant's optimization problem [5]. It is known that for concave games, i.e. concave player utility functions constrained on a convex set, given that the problemb satisfies a constraint qualification condition, then a point satisfying KKT conditions for each player's optimization problem is a Nash equilibrium [152].

We can study the continuous-time dynamical system generated by the gradient of the Lagrangian of each occupant's optimization problem with respect to her own choice variable; we let

$$\dot{x}_i = D_i f_i(x_i, x_{-i}) + \sum_{j \in A_i(x_i)} \mu_{i,j} D_i h_{i,j}(x_i) \tag{2.9}$$

for $i \in \{1, \ldots, n\}$ and where $\mu_{i,j}$ is the j-th dual variable for occupant i's optimization problem. The first term is the derivative of occupant i's utility with respect to her own choice variable x_i. The second term, with the appropriate dual variables $\mu_{i,j}$, ensures that for any initial condition in the feasible set $\mathcal{C}$, the trajectory solving (2.9) remains in $\mathcal{C}$. The right-hand side of (2.9) is the projection of the psuedogradient on the manifold formed by the active constraints at x [152].

We can rewrite the dynamics in a compact form as follows using as an example two convex constraints in the problem. Similar result can be obtained for an arbitrary number of convex constraints. Let $H(x) = [Dh_1 \; Dh_2]$ where $h_j(x) = [h_{1,j} \; \cdots \; h_{n,j}]^T$ for $j \in \{1, 2\}$ and D is the Jacobian operator. Also, let $\mu = [\mu_{1,1} \; \cdots \; \mu_{n,1} \; \mu_{1,2} \; \cdots \; \mu_{n,2}]^T$. Define $F(x, \mu) = \omega(x) + H(x)\mu$.

Then, the dynamics can be written as

$$\dot{x} = F(x, \mu), \;\; \mu \in U(x) \tag{2.10}$$

where

$$U(x) = \left\{ \mu \,\middle|\, \|F(x,\mu)\| = \min_{\substack{\nu_j \geq 0, j \in J(x) \\ \nu_j = 0, j \notin J(x)}} \|F(x,\nu)\| \right\} \tag{2.11}$$

and $J(x) = \{j|\ h_j(x) \leq 0\}$. This formulation is given in the seminal work by Rosen [152] along with the theorem that states that for any initial condition in $\mathcal{C}$, a continuous solution $x(t)$ to (2.10) exists such that $x(t) \in \mathcal{C}$ for all $t > 0$. Thus, we have the following results.

Proposition 2 (Theorem 8 [152])**.** *The dynamical system* (2.10) *is asymptotically stable on* $\mathcal{C}$ *if* $D\omega(x,\mu)$ *has eigenvalues in the open left-half plane for* $x \in \mathcal{C}$ *and* $\mu \in U(x)$*.*

Further, if $x^* \in \mathcal{C}$ is a differential Nash equilibrium, we can linearize ω around x^* and get the following sufficient condition guaranteeing x^* attracts nearby strategies under the gradient flow $F(x,\mu)$.

Proposition 3. *If* $x^* \in \mathcal{C}$ *is a differential Nash equilibrium, and the eigenvalues of* $D\omega(x^*,\mu^*)$ *are in the open left-half plane, then* x^* *is an exponentially stable fixed point of the continuous-time dynamical system* (2.9)*.*

Note that since in our estimation, we restrict $\theta_i \geq 0$, the f_i will be concave; hence, Nash equilibria of the game will be differential Nash equilibria.

These results imply that we can simulate the dynamical system in (2.10) in order to compute Nash equilibria of the game. Using a forward Euler discretization scheme and a sub-gradient projection method, we can compute Nash equilibria of the constrained game. The sub-gradient projection method is known to converge to the unique Nash equilibrium of the constrained n-person concave game [43].

2.2.2 Follower Game: Discrete Choice Games for Agent's Decision-Making Model

In this section, we introduce discrete choice theory for agents' decision-making processes—abstraction. Discrete choice theory is greatly celebrated in the literature as a means of data-driven analysis of human decision-making. Under a discrete choice model, the possible outcome of an agent can be predicted from a given choice set using a variety of available features describing either external parameters or characteristics of the agent.

Let's consider an agent i and the decision-making choice set which is mutually exclusive and exhaustive. The decision-making choice set is indexed by the set $\mathcal{I} = \{\mathcal{J}^1, \ldots, \mathcal{J}^S\}$. Decision maker i chooses between S alternative choices and would earn a **representative utility** f_i for $i \in \mathcal{I}$. Each decision among decision-making choice set leads agents to get the highest possible utility, $f_i > f_j$ for all $i, j \in \mathcal{I}$. In our setting, an agent has a utility which depends on a number of features x_z for $z = 1, \ldots, N$. However, there are several unobserved components—features of the representative utility which should be treated as

random variables. Hence, we define a **random utility** decision-making model for each agent given by

$$\hat{f}_i(x) = g_i(\beta_i, x) + \epsilon_i \tag{2.12}$$

where ϵ_i is the unobserved random component of the agent's utility, $g_i(\beta_i, x)$ is a nonlinear generalization of agent i's utility function, and where

$$x = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N) \in \mathbb{R}^N \tag{2.13}$$

is the collective n features explaining an agent's decision process. The choice of nonlinear mapping g_i and x abstracts the agent's decision. In general, each agent is modeled as a *utility maximizer* that seeks to select $i \in \mathcal{I}$ by optimizing (8.2).

Discrete choice models in their classical representation [173] are given by a linear mapping $g_i(\beta_i, x) = \beta_i^T x$ in which ϵ_i is an independently and identically distributed random value modeled using a Gumbel distribution. According to [173, Chapter 3], the probability that agent i chooses choice $j \in \mathcal{J}$ is given by

$$P_i^j = P(\beta_j^T x + \epsilon_j > \beta_z^T x + \epsilon_z, \forall j \neq z) \Longrightarrow P_i^j = \frac{\exp \beta_j^T x}{\sum_{s=1}^{\mathcal{J}} \exp \beta_z^T x} \tag{2.14}$$

According to (8.4), each agent's probability of a specific choice is given by a Logit model from the linearity assumption for the feature representative utility and the Gumbel distribution modeling the unknown random variable. Other distributions could be used (e.g. Gaussian for Probit model), and this is a design flexibility of discrete choice models.

To model the outcome of the strategic interactions of agents in the generalized gamification abstraction, we use a *sequential non-cooperative discrete game* concept. We enable a more general framework of the discrete choice theory including a temporal dependence term, which supports modeling of stationary or non-stationary data—equilibrium points resulting from sequential equilibrium concept. Introducing a generalized decision-making model for each agent (8.2), in which random utility can be modeled either with linear or nonlinear mapping, a sequential non-cooperative discrete game is given by

Definition 3. *Each agent i has a set $\mathcal{F}_i = f_i^1, \ldots, f_i^N$ of N **random utilities**. Each random utility j has a convex decision-making choice set $\mathcal{I}_j = \{\mathcal{J}_j^1, \ldots, \mathcal{J}_j^S\}$. Given a collective of n features (8.3) comprising the decision process given also the temporal parameter T, agent i faces the following optimization problem for their **aggregated random utilities**:*

$$\max\{\sum_{i=1}^{N} f_i^T(x) |\ f_i \in \mathcal{F}_i\}. \tag{2.15}$$

In the sequential equilibrium concept agents in the game independently co-optimize their aggregated random utilities (8.5) given a collective of n features (8.3) at each time instance

T (temporal parameter). A general incentive design mechanism can motivate their potential actions across various given decision-making choice sets.

The above definition extends the definition of a discrete choice model [173] to sequential games in which agents concurrently co-optimize several discrete (usually mutually exclusive) choices over temporal dependencies. Using this definition, we can apply the proposed game theoretic model by allowing several machine learning algorithms to be directly applied. Machine learning algorithms can potentially be used for modeling the choice of nonlinear mapping (8.2). In particular, Deep Learning models can perform an end-to-end training for higher predictive accuracy using several mini-batched collectives of features (8.3).

2.3 Leader Optimization Problem: Incentive Design

A reverse Stackelberg game is a hierarchical control problem in which sequential decision-making occurs; in particular, there is a leader that announces a mapping of the follower's decision space into the leader's decision space, after which the follower determines his optimal decision [58]. A leader-agent problem occurs when the leader interacts with the agent to perform a task, but the agent is not incentivized to act in the leader's best interests. This conflict is often the result of asymmetric information between the leader and the follower or a disconnect between their goals and objectives.

Specifically in the generalized gamification abstraction towards smart-building energy efficiency (figure 2.1) as followers play in a non-cooperative game, there are sequentially incentive mechanisms announcements $\gamma_{i=1}^{n}$. The leader then observes the equilibrium points—responses of the followers, called $x_{i=1}^{n}$. At the beginning leader updates—estimates the unknown parameters of the followers utility—cost functions. Then, having the estimated parameters, leader solves an optimization problem—maximize its pay-off and design the incentives for the next game round (adaptive incentive design). Overall goal is the convergence of followers actions—responses to leader's desired value.

Both the leader and the followers wish to maximize their pay-off determined by the functions $f_L(x, y)$ and $\{f_1(x, \gamma(x)), \ldots, f_n(x, \gamma(x))\}$ respectively where we now consider each of the follower's utility functions to be a function of the incentive mechanism $\gamma : x \mapsto y$ where leader's decision is $y \subset \mathbb{R}^m$. y can be the default settings for occupants, price—incentives for the followers of the game. The followers' decisions is denoted by x while the leader's strategy is γ.

The basic approach to solving the reversed Stackelberg game is as follows. Let y and x take values in $Y \subset \mathbb{R}^m$ and $\mathcal{C}_i \subset \mathbb{R}$, respectively and let $f_L, f_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for each $i \in \{1, \ldots, n\}$. We define the desired choice for the leader as

$$(x^*, y^*) \in \arg\max_{x,y} \{ f_L(x, y) |\ y \in Y, x \in \mathcal{C} \}. \tag{2.16}$$

Of course, if f_L is concave and $Y \times \mathcal{C}$ is convex, then the desired solution is unique. We have to comment that there are other mechanism design approaches as Vickrey-Clarke-Groves mechanism [119, 131] with some applications in the energy domain [159].

2.3.1 Incentive Design Under Nash

Given that the followers' game is approximated as a Nash game, the incentive problem can be stated as follows:

Problem 1. *Find* $\gamma : X \to Y$, $\gamma \in \Gamma$ *such that* x^* *is a differential Nash equilibrium of the follower game* $(f_1, \ldots, f_n)$ *subject to constraints and* $\gamma(x^*) = y^*$ *where* Γ *is the set of admissible incentive mechanisms.*

By insuring that the desired agent action x^* is a non-degenerate differential Nash equilibrium ensures structural stability of equilibrium helping to make the solution robust to measurement and environmental noise [143]. Further, it insures that it is (locally) isolated — it is globally isolated if the followers' game is concave. $\gamma(x^*)$ is an option provided to the followers and in a sense, it is the outside option. Thus, the leader only selects the constants $y \subset \mathbb{R}^m$. This reduces the solution of the reversed Stackelberg game to a bi-level optimization problem that we solve with a particle swarm optimization (PSO) technique (see, e.g., [90, 91, 176]).

The particle swarm optimization method is a population based stochastic optimization technique in which the algorithm is initialized with a *population* of random solutions and searches for optima by updating *generations*. The potential solutions are called *particles*. Each particle stores its coordinates in the problems space which are associated with the best solution achieved up to the current time. The best over all particles is also stored and at each iteration the algorithm updates the particles' velocities.

At the inner level of the bi-level optimization problem, we replace the condition that the occupants play a Nash equilibrium with the dynamical system determined by the gradients of each player's utility with respect to their own choice variable, i.e.

$$\dot{x}_i = D_i f_i(x_i, x_{-i}, y),\ x_i \in \mathcal{C}_i,\ \forall\ i \in \{1, \ldots, n\}. \tag{2.17}$$

It has been show that by using a projected gradient descent method for computing stationary points of the dynamical system in (2.17), which is derived from an n-person concave games on convex strategy spaces, converges to Nash equilibria [43]. In our simulations, we add the constraint to the leader's optimization problem that at the stationary points of this dynamical system, i.e. the Nash equilibria, the matrix $-D\omega$ is positive definite thereby ensuring that each of the equilibria are non-degenerate and hence, isolated. Then the leader's optimization problem is stated as follows:

Denote the set of non-degenerate stationary points of the dynamical system $\dot{x}$ as defined in (2.17) as $\mathrm{Stat}(\dot{x})$. The leader then solves the following problem: given the joint distribution across agents' states like *active*, *present*, *absent*, find

$$\begin{aligned} &\max_{y \in Y} f_L(y, x) \\ &\text{s.t. } x \in \mathrm{Stat}(\dot{x}) \end{aligned} \tag{2.18}$$

For each particle in the PSO algorithm, we sample from the distribution across player states and compute Nash for the resulting game via simulation of the dynamical system (2.17). Leader can periodically solve (2.18) while updating estimations for followers utility functions. An interesting point is that Nash equilibria estimation and simulation is a robust non-linear dynamic process [131] which resolves much of the uncertainty in the inference process of the utility—parameter estimation under the proposed bi-level optimization problem. Moreover, regarding the necessary size of the data process that will be necessary for the accurate estimation of the utility function we can follow [85]. Under this lower bound formulation the leader could involve the necessary size of data points resulting from followers' approximated Nash for the utility estimation. This will result to more efficient—robust utility estimations for our gamification abstraction (figure 2.1).

2.3.2 Incentive Design Under Discrete Choice Games

If the followers' game is now approximated as a discrete choice game, the incentive problem can be stated as follows:

Problem 2. *Find* $\gamma : X \to Y$, $\gamma \in \Gamma$ *such that* x_τ^* *is an equilibrium of the follower—discrete choice game* $(f_1, \ldots, f_n)$ *and* $\gamma(x_\tau^*) = y^*$ *where* Γ *is the set of admissible incentive mechanisms.*

We have to point out that x_τ^* is an aggregated, possible in time τ, equilibrium of the follower—discrete choice game. As x^* (equilibrium of the follower—discrete choice game) is a discrete value, we transform it in a continuous value x_τ^* and then leader can solve a convex problem. As x_τ^* is an equilibrium of the follower—discrete choice game and $\gamma(x_\tau^*)$ is an option provided to the followers and in a sense. Leader essentially only selects the constants $y \subset \mathbb{R}^m$ and the overall solution of the reversed Stackelberg game is a bi-level optimization problem that we solve with a particle swarm optimization (PSO) technique (see, e.g., [90, 91, 176]).

As in the Nash play incentive design, the inner level of the bi-level optimization problem, we use the dynamical system (solving a set of equations for each player) determined by the gradients of each player's utility given the temporal parameter T, i.e.

$$\dot{x}_i = D_i f_i^T(x_i, y), \; x_i \in \mathcal{C}_i, \; \forall \; i \in \{1, \ldots, n\}. \tag{2.19}$$

Denote the set of non-degenerate stationary points of the dynamical system $\dot{x}$ as defined in (2.19) as $\text{Stat}(\dot{x})$. The leader then solves the following problem: given the joint distribution across agents' states like *active*, *present*, *absent*, find

$$\begin{aligned} &\max_{y \in Y} f_L(y, x_\tau) \\ &\text{s.t. } x_\tau \in \text{Stat}(\dot{x}) \end{aligned} \tag{2.20}$$

2.4 Chapter Summary

This chapter addressed the challenge of formulating a generalized gamification abstraction towards smart-building energy efficiency (figure 2.1). We modeled building occupants' and manager's interaction as a reversed Stackelberg game (leader—followers) in which there are multiple followers (building occupants) that play in a non-cooperative game. As a result the proposed game theoretic general abstraction accounts for games under Nash strategy and discrete choice games. The incentive design supports a bi-level optimization which concurrently maximizes leader's and followers' cost—utility functions. As we will introduce in later chapters, in order to achieve this goal, the leader implements a *social game* in which the followers are pitted against one another. The occupants win points based on their energy consumption choices. These points are then used to determine the individual follower's chance at winning in a lottery. Leader's task is to simultaneously update the incentive feedback and accurately account for the occupants utility parameters estimation towards more and more efficient behavior prediction.

Chapter 3

Data-Driven Utility Learning in Games: A Nash Approach

The main contributions of this chapter are to introduce a base utility estimation framework given a set of non-cooperative, self-interested agents in a continuous game and characterizing their play using the Nash equilibrium concept. Game theoretic analysis often relies on the assumption that the utility function of each agent is known a priori; however, this assumption usually does not hold in many real-world applications in which agents' utilities should be learned given past historic data—agents' strategic actions. Having a gamification content in which players are rational, we want to introduce a generalized framework to answer the following questions:

Can we infer non-cooperative agents' utility functions, which represent accurately their decision-making model, while observing sequential equilibrium points at a non-cooperative continuous game? Is the estimated model for agents' behavior a good predictor of their actual behavior?

Given a set of non-cooperative agents in a continuous game and characterizing their play using the Nash equilibrium concept, we develop a learning scheme for the parameters characterizing their utility functions. Parameterizing their utility functions we use a convex optimization problem to estimate the parameters. We simulate the game defined by the estimated utility functions. For the formulation of the utility estimation problem as a convex optimization problem we use first-order necessary conditions for Nash equilibria. In particular, the gradient of each occupant's utility function should be identically zero at the observed Nash equilibrium. This is the case since the observed Nash equilibria are all inside the feasible region so that none of the constraints are active, i.e. we do not have to check the derivative of Lagrangian of each agent's optimization problem.

3.1 Utility Learning Framework via Inverse Optimization

We fold the utility learning problem using concurrently statistical methods and inverse optimization techniques that serve the creation of a basis utility estimation framework. As we described in chapter 2, our utility estimation scheme fits into the overall incentive design framework introduced in the gamification abstraction towards smart-building energy efficiency (figure 2.1). This motivates why we are interested in learning more than a simple predictive model for agents, but rather an utility-based forecasting framework that accounts for individual preferences. The utility learning framework we propose is quite broad in that it encompasses a wide class of continuous games.

Let's consider a non-cooperative game in which agents' utility functions are following the follower game—Nash approach (2.1). More specifically, we define the ith agent's parametrized utility function $f_i(x_i, x_{-i})$ using parameters $\theta_i = (\theta_{i1}, \ldots, \theta_{im_i}) \in \mathbb{R}^{m_i}$ and a finite set of basis functions $\{\phi_{ij}(x_i, x_{-i})\}_{j=1}^{m_i}$.

Hence, the utility function is such that:

$$f_i(x; \theta_i) = \langle \phi_i(x_i, x_{-i}), \theta_i \rangle + \bar{f}_i(x) \tag{3.1}$$

where $\phi_i = [\phi_{i,1} \ \cdots \ \phi_{i,m_i}]^\top$ and $\bar{f}_i(x)$ is a function that captures *a priori* knowledge of the agent's utility function (e.g., the incentive component designed by the leader). It is important to note that the finite set of basis functions may need to be concave as the simulated N-person game to guarantee a unique Nash equilibrium [152]. In [14] a parametric and non-parametric data-driven approach using theory of variational inequalities is introduced for game theoretic and transportation science models. However, under this approach the notion of unique Nash isn't clear, with introduced uncertainty in the simulation of the Nash—estimated utility functions due to myopic play. In our proposed inverse optimization framework, using a predefined finite set of basis functions, there is a clear notion of unique Nash and robustness at the simulated game.

3.2 Base Utility Estimation Framework

We start by describing the basic utility estimation framework using equilibrium conditions for the game played between the agents. The utility learning framework we propose is quite broad in that it encompasses a wide class of continuous games. At the core of our approach we present that the utility learning problem can be formulated as a convex optimization problem by using first– and second–order conditions for Nash equilibria. This base utility estimation framework will serve as the basis for the robust utility learning method 5.

Each observation $x^{(k)}$ from an agent is assumed to be an ε–approximate differential Nash equilibrium where the superscript notation $(\cdot)^{(k)}$ indicates the k–th observation. For each

observation $x^{(k)}$, it may be the case that only a subset of the agents, say $\mathcal{S}^k \subset \mathcal{I}$ at observation k, participate in the game.

Then notationally each observation is such that

$$x^{(k)} = \left(x_j^{(k)}\right)_{j \in \mathcal{S}^k}. \tag{3.2}$$

If agent i participates in n_i instances of the game, then there are n_i observations for that agent. Let $n = \sum_{i=1}^{p} n_i$ be the total number of observations.

We can consider first–order optimality conditions for each agent's optimization problem and define a residual function capturing the degree of suboptimality of $x_i^{(k)}$ [92, 148]. Indeed, for agent i's optimization problem, define the residual of the stationarity condition to be

$$r_{\mathrm{s},i}^{(k)}(\theta_i, \mu_i) = D_i f_i(x_i^{(k)}, x_{-i}^{(k)}) + \textstyle\sum_{j=1}^{\ell_i} \mu_i^j D_i h_{i,j}(x_i^{(k)}) \tag{3.3}$$

and the residual of the complementary conditions to be

$$r_{\mathrm{c},i}^{j,(k)}(\mu) = \mu_i^j h_{i,j}(x_i^{(k)}),\ j \in \{1, \ldots, \ell_i\}. \tag{3.4}$$

Define

$$r_{\mathrm{c},i}^{(k)}(\mu_i) = [r_{\mathrm{c},i}^{1,(k)}(\mu_i)\ \cdots\ r_{\mathrm{c},i}^{\ell_i,(k)}(\mu_i)]. \tag{3.5}$$

Using data from the agents' decisions (e.g., lighting votes from the social game experiment which we describe in Chapter 4), the base utility learning framework consists of solving the optimization problem given by

$$\begin{aligned} &\min_{\mu,\theta} \textstyle\sum_{i=1}^{p} \sum_{k=1}^{n_i} \chi_i(r_{\mathrm{s},i}^{(k)}(\theta, \mu), r_{\mathrm{c},i}^{(k)}(\mu)) \\ &\text{s.t.}\ \ \theta_i \in \Theta_i, \mu_i \geq 0\ \ \forall\ i \in \{1, \ldots, p\} \end{aligned} \tag{P}$$

where Θ_i is a constraint set on the parameters θ_i that captures prior information about the objective, $\chi : \mathbb{R}^p \times \mathbb{R}^{\sum_{i=1}^{p} \ell_i} \to \mathbb{R}_+$ is a non-negative, convex penalty function satisfying $\chi(z_1, z_2) = 0$ if and only if $z_1 = 0$ and $z_2 = 0$, i.e. any norm on $\mathbb{R}^p \times \mathbb{R}^{\sum_{i=1}^{p} \ell_i}$, and the inequality $\mu_i \geq 0$ is element-wise.

The goal of this optimization problem—which is a finite dimensional optimization problem in the θ_i's—is to find θ_i for each agent such that $(\hat{f}_i)_{i \in \mathcal{I}}$ is consistent (or approximately consistent) with the data. As is noted in [92], we also remark that it is important that the sets Θ_i contain enough prior information about the objectives f_i in order to prevent trivial solutions. For example, if it is the case that $\bar{f}_i(x^{(k)}) = 0$ for each k and each $\Theta_i = \mathbb{R}^{m_i}$ then the trivial solution $\theta_i = \mathbf{0}_{m_i}$ is feasible. For many applications some *a priori* knowledge on part of the utility functions of agents may be encoded in each Θ_i (e.g., choosing Θ_i such that $\theta_{1i} = 1$ or similarly selecting the incentive component of the utility, a design possibility for the leader [147]) or through other normalization techniques to prevent such trivial solutions.

In the context of the social game application, introduced in Chapter 4), we explicitly discuss how to construct this constraint set in such a way that we ensure the estimated utility functions are concave which in turn guarantees that there exists a Nash equilibrium to the estimated game. Hence, it is a modeling and (or) optimization based design technique for the selection of θ_i for each agent.

3.2.1 Constrained Ordinary Least Squares Formulation

After defined the main optimization problem P for the base utility estimation framework, we know transform the problem in a classical statistical model, constrained ordinary least squares (cOLS).

Let's define

$$X_i^{(k)} = \begin{bmatrix} D_i h_i(x_i^{(k)}) & D_i\phi_i(x^{(k)})) \\ \hat{h}_i(x_i^{(k)}) & \mathbf{0}_{\ell_i \times m_i} \end{bmatrix}, \tag{3.6}$$

where

$$\hat{h}_i(x_i) = \operatorname{diag}(h_{i,1}(x_i), \ldots, h_{i,\ell_i}(x_i)), \tag{3.7}$$

$$D_i h_i(x_i) = [D_i h_{i,1}(x_i) \ \cdots \ D_i h_{i,\ell_i}(x_i)], \tag{3.8}$$

and $n_d = (\ell_i + 1)n$ is the total number of data points.

The regressor matrix is then defined as $X = \operatorname{diag}(X_1, \cdots, X_p) \in \mathbb{R}^{n_d \times (\ell_i+1)p}$ where $X_i = [(X_i^{(1)})^\top \ \cdots \ (X_i^{(n_i)})^\top]^\top$.

Define the regression coefficient

$$\beta = [\mu_1^1 \ \ldots \ \mu_1^{\ell_1} \ \theta_1 \ \cdots \ \mu_p^1 \ \ldots \ \mu_p^{\ell_p} \ \theta_p]^\top \in \mathbb{R}^{(\ell_i+1)p} \tag{3.9}$$

and the observation matrix $Y = [Y_1 \ \cdots \ Y_p]^\top \in \mathbb{R}^{(\ell_i+1)p}$ where

$$Y_i = [\bar{f}_i(x^{(1)}) \ \mathbf{0}_{\ell_i} \ \cdots \ \bar{f}_i(x^{(n_i)}) \ \mathbf{0}_{\ell_i}]^\top. \tag{3.10}$$

Using the Euclidean norm for χ in (P) leads to an cOLS problem. Indeed,

$$\min_{\beta} \left\{ \|Y - X\beta\|_2 \middle| \ \beta \in \mathcal{B} \right\} \tag{P1}$$

where $\mathcal{B} = \{\beta |\ \theta_i \in \Theta_i, \mu_i \geq 0, \ \forall i \in \mathcal{I}\}$.

Enforcing that each of the constraint sets Θ_i is encoded by inequalities on θ_i, the above stated problem can be viewed as a classical multiple linear regression model with inequality constraints described by the data generation process

$$Y = X\beta + \epsilon, \ \beta \in \mathcal{B} \tag{3.11}$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_p)$ is the error term satisfying:

- $E(\epsilon|X) = 0^{n_d \times 1}$;
- $\text{cov}(\epsilon|X) = \sigma^2 I^{n_d \times n_d}$;
- $\{\epsilon_i\}_{i=1}^{p}$ independent and identically distributed (iid) with a zero mean and σ^2 variance.

Error term that satisfies the above conditions lies in the assumptions of homoscedasticity. The core idea is that if the above assumptions hold, then the linear estimator satisfies the Markov BLUE property—best linear unbiased estimator. However, in later chapters we will introduce a more generalized constrained model which works under heteroskedasticity.

3.3 Chapter Summary

Using tools from optimization theory and statistical learning theory, we developed a base utility estimation framework with which we infer the utility—cost functions of non-cooperative agents. We refer our approach as a classical constrained ordinary least squares problem with homoscedasticity assumptions for the noise terms. However, there is the general question regarding the accuracy of the proposed model—technique. In the next Chapter 4 we will first introduce an adversarial setting which potential could "attack" our base utility inference. Later, in Chapter 5 we will develop an inference framework accounting for heteroskedasticity in the error terms and we will compare its sequential step ahead predictions with the base utility inference model. Surprisingly, the forecasting accuracy of the base utility inference model is significantly reduced.

Chapter 4

Poisoning Attacks on Utility Learning in Games

Just as many other data-driven methods, utility learning also suffers from potential security risks. Due to the great economic value of accurate forecasting of agents' behaviors, there are huge incentives for adversaries to attack utility learning methods by poisoning training datasets and mislead predictions to achieve malicious goals. In this Chapter, we introduce and analyze optimal poisoning attack strategies in order to understand adversarial actions and further encourage potential defenses. Moreover, we study how an adversary might disguise the attacks by mimicking normal actions. Using a gamification approach and an inverse learning of agents' cost—utility functions, in the current Chapter we want to answer the following questions:

Can adversaries attack an utility learning procedure, as introduced in Chapter 3, and what are the implications of such an attack? How accurately an attacker can mimic normal actions of an agent's decision-making model under a non-cooperative game?

As game theory deals with predicting the behavior of an agent, it can well applied to inform economic and engineering decisions under a Nash game. Usually in game-theoretic analysis, it is assumed that there exists at the best a prior on the model of an individual agent's utility. However, the model primitives are not directly observable and can be difficult to estimate. Small errors in the estimates may substantially affect the predictions of agents' behaviors. Thus, it is crucial to develop accurate estimates of utility models 3.

However, utility learning is a statistical—machine learning technique in which the increasing reliance upon data comes with the security risks. Recent research in machine learning community has shown the vulnerability of data-driven algorithms in various applications [39, 53, 112, 117]. Adding a few well-crafted data points into the training dataset (poisoning attack) or manipulate the testing data (evasion attack) can induce spurious model outputs. These datasets are often crowdsourced or collected from wireless sensor networks,

and therefore easy to falsify. High-profile attacks on spam filtering [113], recommendation system [114], face recognition [17] among others have diminished the credibility of data-driven technologies. There are examples related to smart grid security and attacks on state estimation [75, 76]. In parallel to the study of attack and defensive strategies in the machine learning community, this paper presents the first effort to investigate the effectiveness of data-driven models used for game-theoretical analysis under adversarial conditions.

In the machine learning and statistics communities, earliest treatments consider the robustness of learning to noise, including the extension of the PAC model by Kearns and Li [89], as well as work on robust statistics [22, 62, 175, 184]. In adversarial settings, robust methods for dealing with arbitrary corruptions of data have been proposed in the context of linear regression [26], and high-dimensional sparse regression [25]. These methods are based on assumptions on training data such as sub-Gaussian distribution and independent features. Biggio et al. pioneered the research of optimizing malicious data-driven poisoning attacks for kernel-based learning algorithms such as SVM [16]. The key optimization technique is to approximately compute implicit gradients of the solution of an optimization problem based on first-order KKT conditions. Similar techniques were later generalized to optimize data poisoning attacks for several other important learning algorithms, such as topic modeling [126], and autoregressive models [2].

In unsupervised settings, [153] examined how an attacker can systematically inject traffic to mislead a PCA anomaly detection system for DoS attacks. [95] demonstrated *boiling frog attacks* on centroid anomaly detection that involve incremental contamination of systems using retraining. Theoretical online centroid anomaly detection analysis has been discussed in [95]. Ciocarlie et al. [32] discuss sanitiziation methods against time-based anomaly detectors in which multiple micro-models are built and compared over time to identify poisoned data. The assumption in their system is that the attacker only controls data generated during a limited time window.

In the gamification—game theoretic setting, forecasting agents' behaviors is of paramount importance in various markets, including commodity, energy [99, 101, 103], and ride-sharing systems [146] among others. For example, predicting volatility is the basis for pricing and better forecast results in better returns. Due to the great economic impact of behavioral forecasts, there are huge incentives for adversaries to generate attacks to mislead prediction systems. In the smart building setting attacks in the utility learning scheme (figure 2.1 in Chapter 2) would potentially compromise the efficient learning of building occupants' preferences (followers), which would, in turn, result in inaccurate energy prediction and put the building in a disadvantageous position in the energy markets [81, 82]. Our motivation is to investigate the vulnerabilities of utility learning models by proposing several poisoning attack strategies. We hope to encourage potential defenses via careful adversarial risk analysis.

The contributions of this Chapter—adversarial attack framework are listed as follows:

- We present a general framework to estimate utility functions of individual agents in a non-cooperative game from their historical actions.

- We analyze the optimal poisoning attack strategy and develop an algorithm based on Projected Gradient Ascent to solve for the optimal attack.
- We develop an algorithm to synthesize malicious attack points that imitate normal behaviors and are thereby hard for a defender to detect.
- We demonstrate through experiments on both synthetic and real-world datasets that the proposed poisoning attack strategy outperforms other baseline attack methods.

4.1 Poisoning Utility Learning

Motivated by the utility learning formulation introduced at Chapter 3 we describe the data poisoning attack model and a practical algorithm to solve for optimal attack strategies. This will give several insights about attacks in a gamification setting. Overall utility learning formulation is introduced and defined in Chapter 3. Especially in Section 3.2.1 we consider the squared loss for utility learning. We define

$$X_i^{(k)} = \begin{bmatrix} D_i h_i(x_i^{(k)}) & D_i \phi_i(x^{(k)}) \\ \hat{h}_i(x_i^{(k)}) & \mathbf{0}_{l_i \times m_i} \end{bmatrix} \tag{4.1}$$

where

$$\hat{h}_i(x_i^{(k)}) = \text{diag}(h_{i,1}(x_i^{(k)}), \cdots, h_{i,l_i}(x_i^{(k)})) \tag{4.2}$$
$$D_i h_i(x_i) = [D_i h_{i,1}(x_i), \cdots, D_i h_{i,l_i}(x_i)] \tag{4.3}$$

Further, let $\beta_i = [\mu_{i,1}, \cdots, \mu_{i,l_i}, \theta_i^\top]^\top$ and define $X_i = [(X_i^{(1)})^\top, \cdots, (X_i^{(p_i)})^\top]^\top$ and

$$Y_i = [-D_i \bar{f}_i(x^{(1)}), \mathbf{0}_{l_i}, \cdots, -D_i \bar{f}_i(x^{(p_i)}), \mathbf{0}_{l_i}] \tag{4.4}$$

Hence, we can convert the utility learning problem into a simple constrained ordinary least squares (cOLS) form

$$\min_{\beta_i \in \mathcal{B}_i} \|Y_i - X_i \beta_i\|_2^2 \triangleq \mathcal{L}_{trn}(\mathcal{D}) \tag{4.5}$$

where $\mathcal{B}_i = \{\beta_i | \mu_i \geq 0, \theta_i \geq d_i\}$ and $\mathcal{D} = (x_1^{(k)}, \cdots, x_n^{(k)})_{k=1}^K$ denotes the training dataset. K is the total number of observations in the training set, and $\mathcal{L}_{trn}(\mathcal{D})$ denotes the training loss. The parameters of agent i's utility function, i.e., θ_i, can thus be estimated by the last m_i coordinates of (4.5)'s solution. We can then predict the agents' decisions by simulating the Nash equilibrium of the estimated utility functions.

4.1.1 The Attack Model

The goal of the attacker is to maximally compromise the utility learning algorithm such that the utility functions learned from the observed decisions of agents are barely useful for predicting agents' future decisions in a game. We consider a strong threat model based on the Kerckhoffs's principle [161]. An attacker is a player in the game and attempts to achieve the malicious goal by taking well-crafted actions. We assume that the attacker has the following capabilities:

1. The attacker has access to the training dataset which contains the historical decisions of all agents in the game.
2. The attacker can observe other players' action when taking attack actions
3. The attacker knows the utility learning algorithm

Let a denote the index of the attacker player and $\tilde{x}_a$ denote the attack action. Correspondingly, $\tilde{x}_{-a}$ represents the actions of all non-attacker players during the attack time. The optimal attack strategy can thus be formulated as

$$\max_{\tilde{x}_a} \sum_{k=1}^{K} \sum_{i \in \mathcal{S}^k} (x_i^{(k)} - \hat{x}_i^{(k)})^2 \triangleq \mathcal{W} \tag{4.6}$$

where $x_i^{(k)}$ and $\hat{x}_i^{(k)}$ are the ground truth and the prediction of agent i's k-th decision, respectively. It is worth noting that $\hat{x}_i^{(k)}$ depends on the utility models, which further depend on the attack action $\tilde{x}_a$ because the utility functions are learned by minimizing $\mathcal{L}_{trn}(\mathcal{D} \cup \{\tilde{x}_a, \tilde{x}_{-a}\})$.

4.2 Optimal Attack Strategies

We use the *projected gradient ascent* (PGA) to solve the optimization problem in (4.6). In iteration t, we update $\tilde{x}_a^t$ as follows:

$$\tilde{x}_a^{t+1} = \text{Proj}_{\mathcal{C}_a}(\tilde{x}_a^t + s_t \nabla_{\tilde{x}_a} \mathcal{W}) \tag{4.7}$$

where $\text{Proj}_{\mathcal{C}_a}$ is the projection operator onto the feasible region $\mathcal{C}_a$ and s_t is the step size in iteration t. $\nabla_{\tilde{x}_a} \mathcal{W}$ can be expressed as

$$\nabla_{\tilde{x}_a} \mathcal{W} = -2 \sum_{k=1}^{K} \sum_{i \in \mathcal{S}^k} (x_i^{(k)} - \hat{x}_i^{(k)}) \frac{\partial \hat{x}_i^{(k)}}{\partial \tilde{x}_a} \tag{4.8}$$

Next, we show how to approximately compute $\frac{\partial \hat{x}_i^{(k)}}{\partial \tilde{x}_a}$. This is challenging because two implicit optimization problems are involved to establish the mapping from $\tilde{x}_a$ to $\hat{x}_i^{(k)}$. We apply the chain rule and get

$$\frac{\partial \hat{x}_i^{(k)}}{\partial \tilde{x}_a} = \sum_{j \in \mathcal{S}^k} \frac{\partial \hat{x}_i^{(k)}}{\partial \theta_j} \frac{\partial \theta_j}{\partial \tilde{x}_a} \tag{4.9}$$

We refer to $\frac{\partial \hat{x}_i^{(k)}}{\partial \theta_j}$ as the *testing sensitivity*, which characterizes how the estimated Nash equilibria vary according to the parameters of an agent's utility function. This gradient is non-trivial to evaluate since given θ_j, $\hat{x}_i^{(k)}$ is determined by concurrently maximizing the utility function of each agent who participates in the game. $\frac{\partial \theta_j}{\partial \tilde{x}_a}$ is referred to as the *training sensitivity*, which captures the change of the learned utility models with respect to the attack value $\tilde{x}_a$. The least squares optimization in the training phase dictates how the change of $\tilde{x}_a$ propagates to that of θ_j. Inspired by [16, 114, 183], we leverage the KKT conditions to approximately evaluate the training and testing sensitivity.

4.2.1 Computing the training sensitivity

The KKT conditions of (4.5) states that the optimal solution β_j satisfies

$$2X_j^\top (X_j \beta_j - Y_j) + 2\tilde{X}_j^\top (\tilde{X}_j \beta_j - \tilde{Y}_j) - \nu_j = 0 \tag{4.10}$$

$$\nu_{j,r} \beta_{j,r} = 0, r = 1, \cdots, l_j \tag{4.11}$$

$$\nu_{j,r} (d_{j,r} - \beta_{j,r}) = 0, r = l_j + 1, \cdots, l_j + m_j \tag{4.12}$$

where $\tilde{X}_j$ and $\tilde{Y}_j$ are composed in a similar way to (4.1) and (4.4) using the observed actions of agent j during the attack time, i.e.,

$$\tilde{X}_j = \begin{bmatrix} D_j h_j(\tilde{x}_j) & D_j \phi_j(\tilde{x}_j, \tilde{x}_{-j}) \\ \hat{h}_j(\tilde{x}_j) & \mathbf{0}_{l_j \times m_j} \end{bmatrix} \tag{4.13}$$

and

$$\tilde{Y}_j = \begin{bmatrix} -D_j \bar{f}_j(\tilde{x}_j, \tilde{x}_{-j}) \\ \mathbf{0}_{l_j} \end{bmatrix} \tag{4.14}$$

Note that if agent j is the attacker, then $\tilde{x}_j = \tilde{x}_a$; otherwise, $\tilde{x}_a$ is an element in $\tilde{x}_{-j}$. Among the terms in the KKT conditions, $\tilde{X}_j$, β_j and ν_j are functions of $\tilde{x}_a$, while X_j consists of observed equilibria before the attack time and thereby does not depend on $\tilde{x}_a$.

Assuming the KKT conditions under perturbation of the attack value $\tilde{x}_a$ remain satisfied, we obtain

$$2(X_j^\top X_j + \tilde{X}_j^\top \tilde{X}_j)\frac{\partial \beta_j}{\partial \tilde{x}_a} - \frac{\partial \nu_j}{\partial \tilde{x}_a} = 2\frac{\partial \tilde{X}_j^\top \tilde{Y}_j}{\partial \tilde{x}_a} - 2\frac{\partial \tilde{X}_j^\top \tilde{X}_j}{\partial \tilde{x}_a}\beta_j \tag{4.15}$$

$$\frac{\partial \nu_{j,r}}{\partial \tilde{x}_a}\beta_{j,r} + \nu_{j,r}\frac{\partial \beta_{j,r}}{\partial \tilde{x}_a} = 0, r = 1, \cdots, l_j \tag{4.16}$$

$$\frac{\partial \nu_{j,r}}{\partial \tilde{x}_a}(d_{j,r} - \beta_{j,r}) - \nu_{j,r}\frac{\partial \beta_{j,r}}{\partial \tilde{x}_a} = 0, r = l_j + 1, \cdots, l_j + m_j \tag{4.17}$$

Since $\beta_{j,r}$ and $\nu_{j,r}$ can be computed at each iteration step, we can compute $\frac{\partial \beta_{j,r}}{\partial \tilde{x}_a}$ and $\frac{\partial \nu_{j,r}}{\partial \tilde{x}_a}$ by solving the system of linear equations (4.15)-(4.17), and $\frac{\partial \theta_j}{\partial \tilde{x}_a}$ can be obtained from the last m_j coordinates of $\frac{\partial \beta_j}{\partial \tilde{x}_a}$.

4.2.2 Computing the testing sensitivity

To compute $\frac{\partial \hat{x}_i^{(k)}}{\partial \theta_j}$, we consider the optimality conditions of each agent's optimization problem at the Nash equilibrium. For agent i, we have

$$\langle D_i \phi_i(\hat{x}_i^{(k)}, \hat{x}_{-i}^{(k)}), \theta_i \rangle + D_i \bar{f}_i(\hat{x}^{(k)}) + \sum_{j \in \mathcal{A}_i(\hat{x}_i^{(k)})} \mu_{i,j} D_i h_{i,j}(\hat{x}_i^{(k)}) = 0 \tag{4.18}$$

$$h_{i,j}(\hat{x}_i^{(k)}) = 0, \forall j \in \mathcal{A}_i(\hat{x}_i^{(k)}) \tag{4.19}$$

The equations can be differentiated with respect to $\tilde{x}_a$ using the chain rule and the result of the differentiation is

$$\begin{aligned}
&\langle \sum_{h=1}^{n} D_{i,h}\phi_i(\hat{x}_i^{(k)}, \hat{x}_{-i}^{(k)})\frac{\partial \hat{x}_h^{(k)}}{\partial \theta_j}, \theta_i \rangle + 1(j=i) D_i \phi_i(\hat{x}_i^{(k)}, \hat{x}_{-i}^{(k)}) \\
&+ \sum_{h=1}^{n} D_{i,h}\bar{f}_i(\hat{x}^{(k)})\frac{\partial \hat{x}_h^{(k)}}{\partial \theta_j} + \sum_{j \in \mathcal{A}_i(\hat{x}_i^{(k)})} \frac{\partial \mu_{i,j}}{\partial \theta_j} D_i h_{i,j}(\hat{x}_i^{(k)}) \\
&+ \sum_{j \in \mathcal{A}_i(\hat{x}_i^{(k)})} \mu_{i,j} D_{i,i} h_{i,j}(\hat{x}_i^{(k)})\frac{\partial \hat{x}_i^{(k)}}{\partial \theta_j} = 0
\end{aligned} \tag{4.20}$$

$$D_i h_{i,j}(\hat{x}_i^{(k)})\frac{\partial \hat{x}_i^{(k)}}{\partial \theta_j} = 0, \forall j \in \mathcal{A}_i(\hat{x}_i^{(k)}) \tag{4.21}$$

where $1(\cdot)$ stands for the indicator function and $D_{i,h}$ denotes the second partial derivative with respect to agent i's and h's decision. To solve for $\frac{\partial \hat{x}_i^{(k)}}{\partial \theta_j}$, we can calculate the optimality

conditions for all agents, treat them as a system of linear equations in $\frac{\partial \hat{x}_i^{(k)}}{\partial \theta_j}$ $(i = 1, \cdots, n)$, and compute the solution.

Algorithm 1 summarizes the proposed optimal attack strategy. The algorithm can be easily extended to optimizing multiple attack actions by sequentially adjusting one attack action at a time.

Algorithm 1 Optimizing $\tilde{x}_a$ via PGA

Input : Original training dataset $\mathcal{D}_{trn}$, step size $\{s_t\}_{t=1}^{\infty}$, non-attacker players' actions during the attack time $\tilde{x}_{-a}$

Output: Attack action $\tilde{x}_a$

Initialization: Initialize $\tilde{x}_a^0$ with values uniformly randomly sampled from $\mathcal{C}_a$; $t = 0$

while $\tilde{x}_a^t$ *does not converge* **do**

  $\{\theta_i\}_{i=1}^n \leftarrow$ learn utility functions on $\mathcal{D}_{trn} \cup \{\tilde{x}_a^t, \tilde{x}_{-a}\}$

  $\{\hat{x}_1^{(k)}, \cdots, \hat{x}_n^{(k)}\}_{k=1}^K \leftarrow$ predict agents' decisions using the learned utility functions

  Compute $\nabla_{\tilde{x}_a} \mathcal{W}$ using (4.8)

  Update: $\tilde{x}_a^{t+1} = \text{Proj}_{\mathcal{C}_a}(\tilde{x}_a^t + s_t \nabla_{\tilde{x}_a} \mathcal{W})$

  $t \leftarrow t + 1$

end

4.3 Mimicking Normal Agent Behaviors

Normally, agents would make decisions that maximize their utility. As a result, malicious actions aimed at compromising utility learning will choose some extreme values that can be easily identified by running a simple statistical test against the normal actions. To mitigate this issue, we propose an alternative approach to computing data poisoning attacks such that the resulting malicious actions mimic the normal ones, but at the same time render the utility learning procedure less effective.

We adopt a Bayesian formulation to take into account both data poisoning and detection avoidance objectives. The prior distribution captures the normal decisions and is defined as a Gaussian distribution

$$p_0(\tilde{x}_a) = \mathcal{N}(\tilde{x}_a^*, \sigma_a^2) \tag{4.22}$$

where $\tilde{x}_a^* = arg\max_{\tilde{x}_a} f_a(\tilde{x}_a, \tilde{x}_{-a})$ represents the optimal action that an agent would take in the game in normal cases. σ_a is the standard deviation for the normal actions. The likelihood $p(\mathcal{D}_{trn}|\tilde{x}_a)$ is defined as

$$p(\mathcal{D}_{trn}|\tilde{x}_a) = \frac{1}{Z} \exp(\gamma \mathcal{W}) \tag{4.23}$$

where $\mathcal{W}$ is the training loss defined in (4.6), Z is a normalization constant and $\gamma > 0$ is a parameter that adjusts the tradeoff between attack performance and detection avoidance.

Decreasing γ will shift the posterior of $\tilde{x}_a$ towards its prior, which makes the resulting attack strategy less effective but more difficult to detect, and vice versa.

Given the prior and likelihood function, an attack strategy that is mindful of possible detection can be obtained by sampling from the posterior distribution

$$p(\tilde{x}_a|\mathcal{D}_{trn}) = \frac{p(\mathcal{D}_{trn}|\tilde{x}_a)p_0(\tilde{x}_a)}{p(\mathcal{D}_{trn})} \tag{4.24}$$

$$\propto \exp(-\frac{(\tilde{x}_a - \tilde{x}_a^*)^2}{2\sigma_a^2} + \gamma\mathcal{W}) \tag{4.25}$$

Sampling from the above posterior distribution is intractable due to the complex dependence of $\mathcal{W}$ on $\tilde{x}_a$. To circumvent this problem, we apply Stochastic Gradient Langevin Dynamics (SGLD) [181] to approximately sample the posterior. More specifically, SGLD iteratively computes a sequence of posterior samples $\{\tilde{x}_a^t\}_{t\geq 0}$, and in iteration t the new sample is calculated by

$$\tilde{x}_a^{t+1} = \tilde{x}_a^t + \frac{s_t}{2}\left(\nabla_{\tilde{x}_a} \log p(\tilde{x}_a|\mathcal{D}_{trn})\right) + \epsilon_t \tag{4.26}$$

where $\{s_t\}_{t\geq 0}$ are step sizes and $\epsilon_t \sim \mathcal{N}(\mathbf{0}, s_t\mathbf{I})$ is independent Gaussian noise. The gradient $\nabla_{\tilde{x}_a} \log p(\tilde{x}_a|\mathcal{D}_{trn})$ is given by

$$\nabla_{\tilde{x}_a} \log p(\tilde{x}_a|\mathcal{D}_{trn}) = -\frac{(\tilde{x}_a - \tilde{x}_a^*)}{\sigma_a^2} + \gamma\nabla_{\tilde{x}_a}\mathcal{W} \tag{4.27}$$

where $\nabla_{\tilde{x}_a}\mathcal{W}$ can be computed using (4.8) and the procedure described in Section 4.2.1 and 4.2.2. Finally, the sampled malicious action $\tilde{x}_a^T$ is projected onto the feasible set $\mathcal{C}_a$. The pseudo-code of the proposed method is provided in Algorithm 2.

Algorithm 2 Optimizing $\tilde{x}_a$ via SGLD

Input : Original training dataset $\mathcal{D}_{trn}$, step size $\{s_t\}_{t=1}^{\infty}$, non-attacker players' actions during the attack time $\tilde{x}_{-a}$, tuning parameter γ, the number of SGLD iterations T, the parameter of the attacker's utility function θ_a

Output: Attack action $\tilde{x}_a$

Initialization: Compute the mean and standard deviation of the normal action $\tilde{x}_a^* = arg\max_{\tilde{x}_a} f_a(\tilde{x}_a, \tilde{x}_{-a}; \theta_a)$, σ_a, and sample $\tilde{x}_a^0 \sim \mathcal{N}(\tilde{x}_a^*, \sigma_a^2)$

for $t = 0$ *to* T **do**

  $\{\theta_i\}_{i=1}^n \leftarrow$ learn utility functions on $\mathcal{D}_{trn} \cup \{\tilde{x}_a^t, \tilde{x}_{-a}\}$

  $\{\hat{x}_1^{(k)}, \cdots, \hat{x}_n^{(k)}\}_{k=1}^K \leftarrow$ predict agents' decisions using the learned utility functions

  Compute $\nabla_{\tilde{x}_a}\mathcal{W}$ using (4.8)

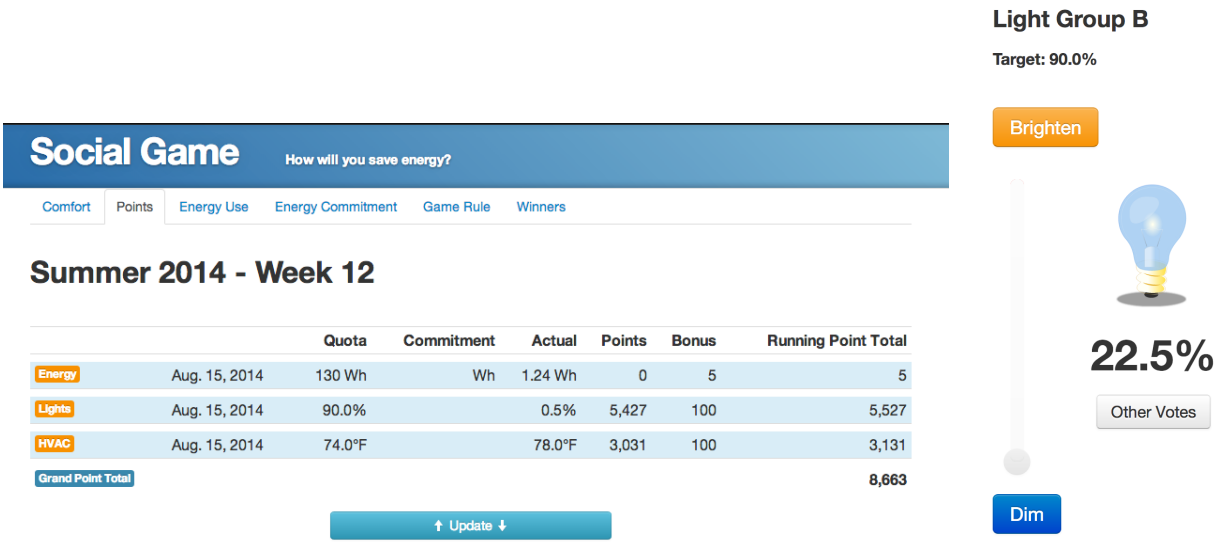
  Update $\tilde{x}_a^{t+1}$ according to (4.26)

end

Project $\tilde{x}_a^T$ onto $\mathcal{C}_a$

4.4 Application To Smart Building Social Game

We evaluate the efficacy of the proposed attack strategies on both synthetic and real-world experimental data collected from a smart building social—energy game.



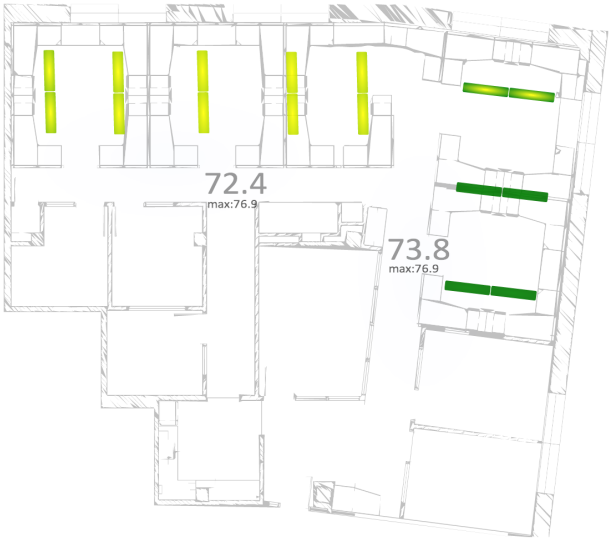
(a) (b)

Figure 4.1: Graphical user interface (GUI) for energy based social game: (a) Display, in table form, of points and votes for energy consumption, HVAC, and lights. (b) Display of the GUI for logging lighting setting preferences.

4.4.1 Social Game Experimental Set-Up

Our experimental setup is in a collaboratory space—an open, shared work space with cubicles—within the CREST center on the UC Berkeley. We crafted a social game such that occupants in this collaboratory freely vote according to their usage preferences of shared resources and are rewarded with points based on how energy efficient their strategy is in comparison with the other occupants. We employ a lottery mechanism consisting of three Amazon gift cards executed bi-weekly to reward occupants; occupants with more points are more likely to win the lottery.

The office is divided into five lighting zones and two heating, ventilating, and air conditioning (HVAC) zones. In this space, there is a total 20 occupants (not randomly selected) who are eligible to participate in the social game. If the occupants are not present in the office, they are excluded from the game at that time instant. When they arrive at the office, they can rejoin the game. To enforce the rule that those who are not present in the space cannot vote remotely, we executed a simple presence detection algorithm based on their



(a)

(b)

Figure 4.2: Occupants can access a variety of information when they log into the social game portal, including various displays of energy consumption by other participants in the game: (a) Display of current light level and temperature in the collaboratory space; energy efficiency of the lights is coded by color where light green indicates *higher energy efficiency*. (b) Display of collaboratory floor plan with dots indicating where present and participating players sit. Players not in the office are excluded from the game. The color of the dot indicates the level of energy efficiency of the player as compare to the other participants; green indicates higher efficiency while red indicates lower efficiency.

power usage [84]. Social game platform and Django (python based) implementation can leverage other occupancy sensing methods using electricity [74], CO2 measurements [83, 86], and sound [71]. Moreover, social game web-portal—platforms can potential be benefited by indoor activity recognition [79, 197, 203] and positioning using maps [68, 194, 195, 198, 200, 202].

We have installed a Lutron[1] system for precise control of the lighting setting (dim level of the lights) in the office as well as desk–level energy monitoring devices (i.e. ACME wireless sensors [72]) to meter the energy usage of each occupant. In addition, we have modified the HVAC system so that it can be precisely controlled. We have verified prior to our experiment that implemented control of these systems results in expected performance.

We have developed a platform to interface with the occupants as well as manage and process collected data. The platform includes a web portal and mobile app that the occupants may use to participate in the game. It also allows for occupants to visualize different aspects of the social game—e.g., the lighting setting and the energy efficiency level of different occupants or the entire building—as well as view the point level and historical voting record of other occupants among many other statistics. Figure 4.1 shows the user interface for viewing points and logging votes. Figure 4.2a shows a visualization of the current light level using a green–to–red color scale with green being more energy efficient. The current temperature is also displayed. Figure 4.2b shows a visualization of each present and participating occupant's energy efficiency level.

Prior to the start of the social game experiment, the lighting setting was 90% of the maximum possible lighting setting. At the start of the social game experiment, we set a default lighting setting which acts as the suggested lighting setting and is the dim level setting in the office if, e.g., no occupants are participating in the game. Throughout the game, we adjust the default lighting setting as well as the points. The lottery mechanism coupled with the points we distribute compose the *incentive component* of the feedback to the participants while the default lighting level is the *physical control component* of the feedback. These two mechanisms act as our control inputs and our feedback mechanism to the participants. We seek to design them by taking into consideration the preferences of the participants. In this way, these mechanisms close the loop around the participant and with our proposed utility learning scheme, these mechanisms can be modified to encourage more energy efficient resource consumption.

The game is designed to leverage interactions amongst occupants, who win points based on how energy efficient their lighting vote is compared to others. An occupant's vote is for the lighting setting in their zone as well as for neighboring zones. The occupants select their desired lighting setting in the continuous interval $[0, 100]$ where each value represents the percentage of the maximum lighting setting possible in the space. The occupants can vote as frequently as they like and the average of all the occupants' current votes sets the implemented lighting setting in the collaboratory. An occupant can leave the lighting setting as the default level after logging in or they can change it depending on their preferences and

[1] http://www.lutron.com/en-US/Pages/default.aspx

other environmental factors that may affect their choice.

The experimental trials reported on in this paper were conducted over the period of 285 days[2]. Experiments with 4 different default levels, $\{20\%, 10\%, 60\%, 90\%\}$, were conducted, covering a spectrum of lighting conditions. Since occupants were allowed to vote whenever they chose, their response rate per day varies. The data set we collected consists of occupant votes (meaning the lighting level they select) over the period of investigation as well as the points that were distributed to each occupant. We collected 6,885 votes over the period of the experiment.

4.4.2 Brief Background: Social Game Experiments

In order to place the work pertaining to building energy efficiency in the context of the state of the art, we briefly overview existing approaches. Recognizing that HVAC systems are responsible for a large portion of building energy consumption, many control theoretic approaches (see, e.g., [6]) derive model predictive and distributed control polices for HVAC systems. While these control theoretic approaches make efforts to account for the presence of occupants, they tend to ignore occupant behaviors and, more importantly, their heterogeneous preferences.

There are other works that make strides towards incorporating behavioral models of occupants; e.g., the authors of [20] employ a multi-agent systems approach to develop a framework for incorporating occupant comfort preferences and the authors of [21] develop behavioral models for lighting usage. In a more active approach, the authors of [166] develop a *collaborative setting definition paradigm* in which occupants and facilities managers submit preferences and requirements and a rule engine tries to resolve them in order to create a universal control policy. While occupants' preferences are taken as inputs to the building control design, it is not clear that it is possible to satisfy all the occupants' comfort preferences simultaneously with those of the facilities manager; hence, the misalignment between preferences and incentives remains. This is a great application of personal comfort models in building design [93].

In our approach, on the other hand, we leverage a social game that creates (friendly) competition between users and employs incentives to resolve conflicting preferences by compensating users. Within the energy application domain, gamification has been largely used for education or awareness (see, e.g., [8, 162]). There are works that are closely related to ours in the sense that they also recognize that occupants are self-interested participants in smart buildings and try to account for their strategic behavior. For example, in [115], the authors develop an interesting scheme for engaging occupants directly in DR. Analogous to our approach, occupants are modeled as utility maximizers in a game theoretic context where they are incentivized to curtail their consumption in response to an event. Our approach differs in that we focus on shared resources such as lighting and HVAC instead of personal devices (e.g., desk appliances). Furthermore, it is assumed in [115] that the type

[2]The period of the experiment was 2014/3/3–2014/12/14.

space (i.e. their preferences) of the users is a known finite set of two possible values. We do not assume the facility manager knows the utility function or the type of the users and we propose an algorithm for learning this utility function from observations of decisions.

While incorporating occupant preferences into building automation is not novel in and of itself, we propose an innovative algorithm for learning occupant preferences in competitive environments and, moreover, learn how their actions are correlated. Such correlations can be leveraged in improving incentive mechanisms to shape users' preferences thereby providing more flexibility. Our method is applied to real-world data from experimental trials we conducted as opposed to simulations as is the case with many existing works. Furthermore, it is agnostic to the application and could be applied in general to other scenarios in which users are competing for constrained but shared resources. For example, the utility learning method can be easily adapted to learning preferences of individual buildings interacting with an aggregator or learning preferences of drivers seeking on-street parking. In each of these cases, there exists a planner—the aggregator or department of transportation—tasked with managing a resource being consumed by self-interested users.

4.4.3 Occupant Decision-Making Model

Each agent's vote x_i is constrained to be in the interval $[0, 100] \subset \mathbb{R}$. Let $\bar{x}$ denote the average of the lighting votes and the setting that is implement—e.g., at observation instance indexed by k, $\bar{x}^{(k)} = \frac{1}{|\mathcal{S}^k|} \sum_{j \in \mathcal{S}^k} x_j^{(k)}$. We model each agent's utility as being composed of two basis functions that capture the tradeoff between desired lighting (satisfaction) and desire to win. The lighting satisfaction an occupant feels may be a function of several factors including their productivity (ability to perform their job) as well as physical comfort. We abstractly model their desired lighting level using a Taguchi loss function, $\psi_i(x_i, x_{-i}) = -(\bar{x} - x_i)^2$, which is interpreted as modeling occupant dissatisfaction in such a way that it is increasing as variation increases from their reported desired lighting setting (their vote) [170].

We acknowledge that an agent may have some internal desired lighting level that is different than its vote; e.g., the agent may realize that voting an extreme value pushes the average toward a more desirable setting. This type of *gaming* results in *moral hazard* type issues which can be addressed in the incentive design step [19, 106]. Thus, we set this type of gaming aside for the time being, and focus instead on the unknown preferences—a different kind of asymmetric information that leads to *adverse selection*—between *lighting* and *winning*.

Points are distributed by the planner using the relationship $\rho(x_b - x_i)(p(x_b - \bar{x}))^{-1}$ where x_b is the baseline setting for the lights. For the experiment $x_b = 90\%$, i.e. the lighting setting used before the implementation of the social game. However, we model each occupant as having a *winning* basis function given by $\phi_i(x_i, x_{-i}) = -\rho c\,(x_i)^2$ where ρ is the total number of points distributed by the planner and c is a scaling factor that is used primarily to scale the two terms of the utility function given that we artificially inflate the points offered in

order to increase their appeal to players and thus induce greater participation[3]. The form of the winning function can be interpreted as capturing the perception that by voting zero, the occupant is selecting the action that will provide the greatest return of points given that points are awarded based on how energy efficient their vote is compared to others[4].

Hence, the utility functions for the social game are modeled as

$$f_i(x_i, x_{-i}) = \theta_i \underbrace{(-\rho c x_i^2)}_{\text{desire to win}} - \underbrace{(\bar{x} - x_i)^2}_{\text{lighting satisfaction}} \tag{4.28}$$

The constraint sets $\mathcal{C}_i$ for each player are determined by the box constraints on the lighting vote for that player, i.e. $\mathcal{C}_i = \{x_i \in \mathbb{R} |\ h_{i,j}(x_i) \geq 0,\ j \in \{1,2\}\}$ where $h_{i,1}(x_i) = 100 - x_i$ and $h_{i,2}(x_i) = x_i$.

In order to formulate (P) for the social game application, we need to determine the admissible parameter sets Θ_i, $i \in \mathcal{I}$ in such a way that we ensure the estimated utility functions are concave and such that equilibria of the estimated game are isolated. We derive a lower bound θ_{LB} such that all $\theta_i \in \Theta_i = \{\theta_i \in \mathbb{R} |\ \theta_i > \theta_{\text{LB}}\}$, $i \in \mathcal{I}$ induce games with these characteristics. To this end, we utilize the second derivative condition on players' utility functions; that is, if for each $i \in \mathcal{I}$, $D^2_{i,i} f_i(x) < 0$, then the game is concave. Computing $D^2_{i,i} f_i$ and using some algebra, we have that $\theta_i > -(c\rho)^{-1}(1 - p^{-1})^2$ where the right-hand side is a negative non-increasing function of p. Thus, concavity is ensured regardless of the number of players by setting $p = 2$, the minimum number of players in a non-cooperative game. Then, given fixed ρ and $0 < \zeta << 1$, the lower bound $\bar{\theta}_{\text{LB}} = -(4c\rho)^{-1} + \zeta$ will guarantee the estimated game is concave.

If $D\omega(x, \mu)$ is invertible, we know that differential Nash equilibria are isolated [144]. Hence, we can augment the constraint sets Θ_i to encode this condition. Given the structure of the utility functions, $D\omega(x, \mu)$ is simply the game Hessian $H = [H_{i,j}]_{j,i=1}^{p}$ with $H_{i,i} = D^2_{i,i} f_i$ and $H_{i,j} = D^2_{i,j} f_i$. Hence, if H is invertible, then the differential Nash are isolated; this is guaranteed for $p \geq 4$ provided the constraint defined by $\bar{\theta}_{\text{LB}} = -(4c\rho)^{-1} + \zeta$ using $\zeta = 10^{-2}$. Indeed, let $H(p)$ denote the game Hessian as a function of the number of players and note that for a particular p, with some simple algebra, it is easy to write $H(p)$ as a off-diagonal matrix constant matrix such that $H_{ii} = d_i + \alpha$ and $H_{i,j} = \alpha$ where $d_i = -2(1 - 1/p) - 2c\rho\theta_i$ and $\alpha = 2(p-1)/p^2$. It is straightforward to verify by determining the eigenvalues of H as

[3]Inflating the points is a process of *framing* [174]—that is, dependent on how the reward system is presented to agents greatly impacts their participation. Framing is routinely used in rewards programs for credit cards among many other point-based programs. The scaling factor c in the winning function removes the framing effect from the estimation procedure. It is selected to ensure the scale of the two basis functions are similar.

[4]We explored other forms of the winning function including the log function, a quasi-concave function that is typically used to represent how individuals value money since it represents the diminishing returns property well [149]. However, the quadratic form of the function we report on here significantly outperformed other choices so that, for the purpose of a prescriptive model, it captures the agents' perceptions about the point distribution mechanism and their value more accurately.

p varies via the method described in [50] that for $p \geq 4$, H will be invertible . For the social game data, at each observation indexed by k, the number of participating players is at least 4. Thus, to ensure concavity and isolated equilibria of the estimated social game, we define $\Theta_i = \{\theta_i \in \mathbb{R} |\ \theta_i > \bar{\theta}_{\text{LB}}\}$ with $\bar{\theta}_{\text{LB}} = -(c\rho 4)^{-1} + \zeta$ with $\zeta = 10^{-2}$.

4.4.4 Evaluation on Synthetic Data

In our first attempt we evaluate the attack strategy on a synthetic dataset. Using the parametric utility functions in (4.28), we devise profiles for eight users with different parameters, and include three default users in order to simulate the real-world game scenario. Default users follow the pre-defined lighting vote without actively adjusting it. Moreover, we randomly select the number of active users in each simulation to simulate the fact that the number of occupants who are present at the office and vote for lighting varies across time in a real game.

We simulate the synthetic data by computing the Nash equilibrium of the agents who are present at the office. The synthetic data is further split into training and testing sets. We consider one of the agents in the game the attacker who aims at hampering the process of utility learning by taking attack actions. Since training data is arranged in chronological order, we assume that the attacks happen at the end of the training period when the attacker can observe the entire history of all agents' actions. To evaluate the impact of attack actions on the utility learning, we examine the predictive power of the utility functions learned from the poisoned training dataset. The training and testing error in terms of root mean square error (RMSE) between the ground truth and predicted actions are used as a measure of the predictive power. Because our goal is to evaluate how well the utility function can predict agents' normal behaviors, the training error is calculated only on the portion of training set without poisoning instances as illustrate in Fig. 4.3.

We compare the PGA strategy with the uniform attack where the poisoning actions are sampled uniformly at random from the allowable lighting vote range $[0, 100]$. We plot the training and testing error against the percentage of poisoning instances for different attack strategies in Fig. 4.4. The prediction errors without attacks are also shown as a baseline. Fig. 4.4 indicates that the training and testing error increase as more poisoning instances are added into the training set. It is clear that PGA is more effective than the uniform attack in reducing the predictive power of the learned utility functions.

4.4.5 Evaluation on Real-World Data

Next, we compare different attack strategies on the dataset collected in a real-world social game lasting 18 days. The dataset contains 19 users' lighting votes recorded every 6 hours. Since the amount of the data is limited, we utilize the entire dataset for learning utility functions. The efficacy of different attack strategies is evaluated based on the prediction error on the pristine part without adversarial instances. Fig. 4.5 shows that PGA produces the largest RMSE compared with the other methods. However, PGA only considers the objective

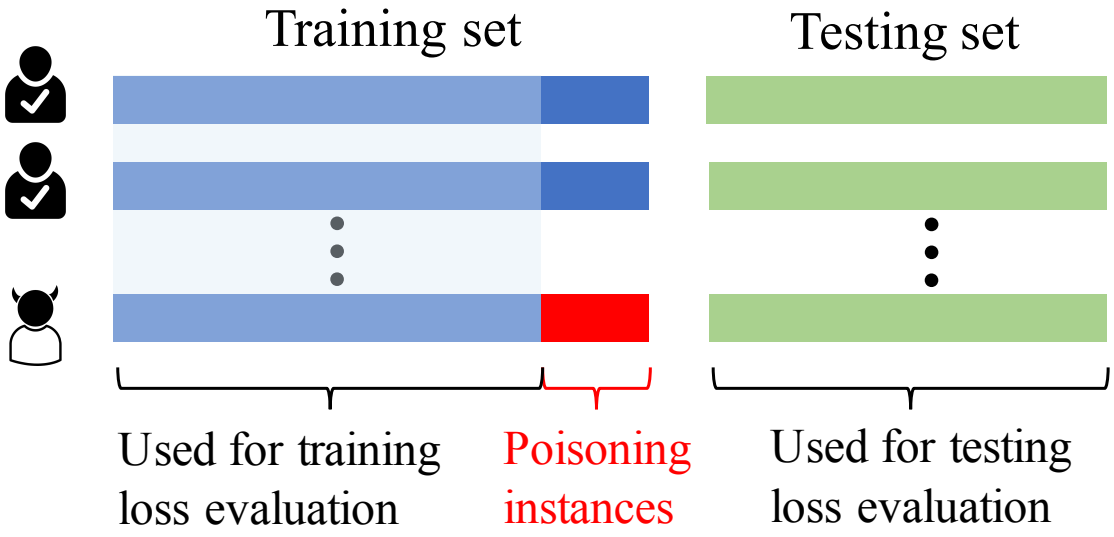
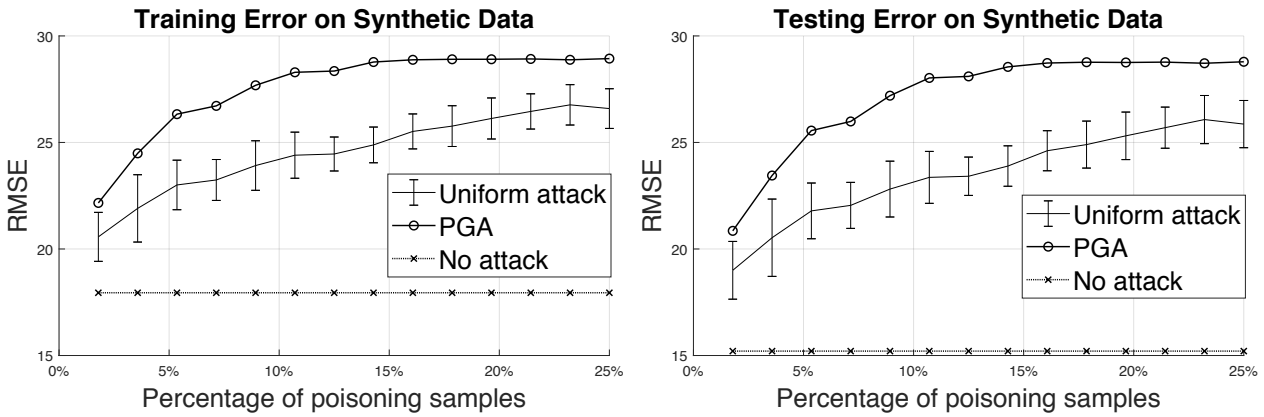


Figure 4.3: Setup of attack effectiveness evaluation.



(a) (b)

Figure 4.4: RMSE for predicting agents' actions using the utility functions learned from the training set with different percentage of poisoning instances.

of compromising the predictive power of the learned utility functions, and therefore might be detected by an informed defender. As shown in Table 4.1, the paired t-test on the poisoning actions produced by PGA rejects the null hypothesis that the poisoning actions produced by the attack strategies are the same as normal actions. In contrast, SGLD has slightly lower attack efficacy but can generate poisoning actions that are difficult to distinguish from the normal actions. The t-test results for SGLD are also presented in Table 4.1. It can be seen that the null hypothesis cannot be rejected when SGLD is used, and increasing the

parameter γ can disguise the poisoning actions better.

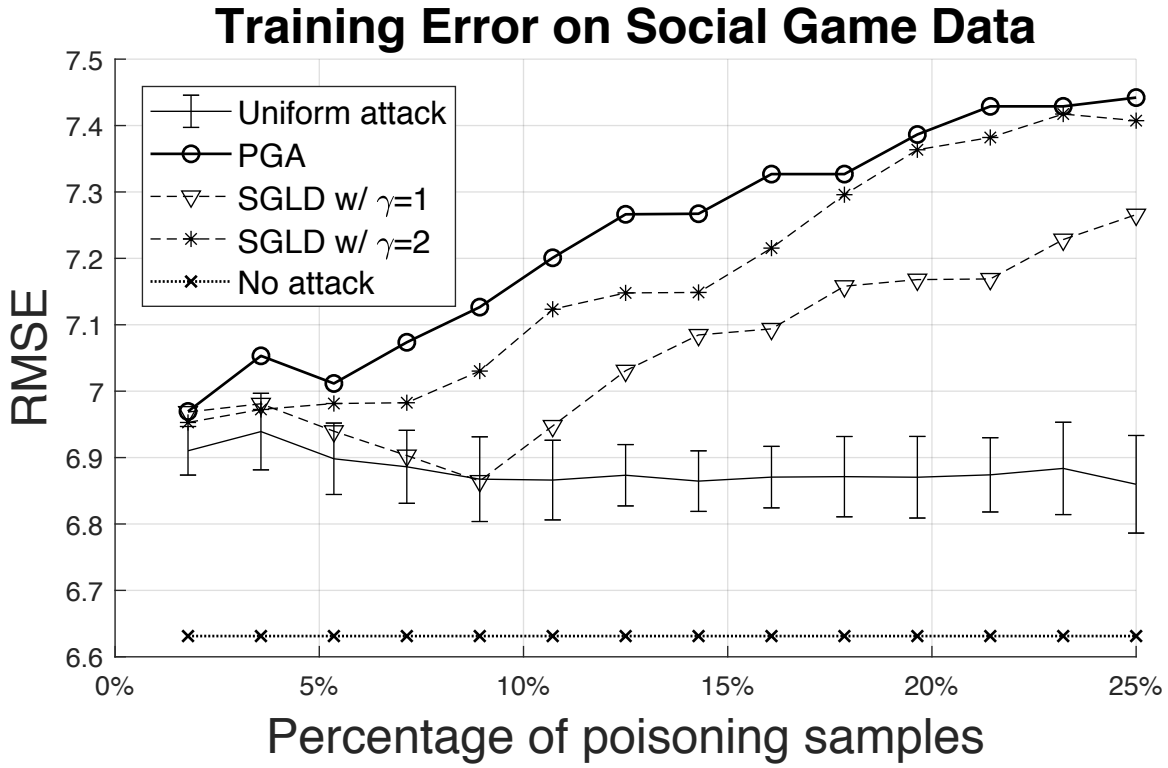


Figure 4.5: Comparing the efficacy of different attack strategies on real-world social energy game data.

Table 4.1: Paired t-test on the null hypothesis that there is no difference in the mean between the distribution of poisoning attack actions and that of normal actions.

Attack strategy	**Test result with 5% significance level**	**P-value**
PGA	Reject the null	0.045
SGLD $\gamma = 1$	**Not reject the null**	0.423
SGLD $\gamma = 2$	**Not reject the null**	0.403

4.5 Chapter Summary

This Chapter studies the vulnerabilities of data-driven utility learning algorithms in adversarial environments. We demonstrate how a powerful attacker can generate malicious poisoning attack instances that give rise to large errors in predicting agents' behaviors but at the same time remain indistinguishable from normal actions. Finally, we conduct extensive experiments on both synthetic and real-world game datasets, and show that the proposed PGA attack strategy is more effective in compromising utility learning compared with other baseline methods. The proposed attack strategies are evaluated on both synthetic and real-world social energy game data, and the results show that the root mean squared error in predicting agents' actions increases by up to 67% by adding only 5% well-crafted poisoning training instances.

Our ultimate goal for the poisoning attack analysis is to develop possible defensive strategies. From the experimental results, there exists a tradeoff between attack efficacy and detectability. Therefore, dynamically tracking deviations in action patterns to detect malicious actions and applying robust learning methods for utility estimation can be potential defenses. Moreover, developed robust learning methods for utility estimation can help boosting the overall forecasting accuracy. In Chapter 5 we will introduce a unified framework for utility estimation in non-cooperative games which also could support potential defenses under given attacks. This proposed robust utility estimation will significantly improve forecasting accuracy and provide insights of agents decision-making.

Chapter 5

Robust Utility Learning Framework via Inverse Optimization

Motivated by Chapters 3 and 4 we propose a generalized robust parametric utility learning framework that employs constrained feasible generalized least squares estimation with heteroskedastic inference. Main goal for this new utility inference technique is to improve forecasting performance, extend the robust utility learning scheme by employing bootstrapping with bagging, bumping, and gradient boosting ensemble methods. The capabilities of such an inference scheme are impressive in the forecasting setting—accuracy. Additionally, it creates a totally different—more complicated learning algorithm which results as a natural defence to poisoning attacks that the base utility learning suffers from. For the quantification of the performance of the proposed robust learning technique we are using occupant voting data for shared resources such as lighting, and we simulate the game defined by the estimated utility functions.

The broader purpose of this Chapter is to present an exceptional inference framework to the proposed generalized gamification abstraction towards smart-building energy efficiency (figure 2.1) that leverages game-theoretic concepts to learn models of players' decision-making in competitive environments (in the smart building setting). More concretely, by modeling decision-making agents as *utility maximizers* and, using inverse optimization and game-theoretic techniques, we derive a robust scheme to infer their utility functions. At the core of our approach is the fact that we model the agents as non-cooperative players in a game playing according to a *Nash equilibrium strategy*. From this point of view, agents are strategic entities that make decisions based on their own preferences despite others. The game-theoretic framework both allows for qualitative insights to be made about the outcome of such selfish behavior—more so than a simple prescriptive model—and, more importantly, can be leveraged in designing mechanisms for incentivizing agents.

Assuming a general parametric form of utility function for each player that is dependent on the decisions of others. Correlations between players' decisions are not known *a priori*, and interestingly this is the core problem that makes base utility learning scheme 3 to have limited inference capabilities. Assuming observations are approximately Nash equilibria, we

use first–and second–order conditions on player utility functions to construct a constrained regression model. The result is as a constrained Generalized Least Squares (cGLS) problem with non-spherical noise error terms. Using constrained Feasible Generalized Least Squares (cFGLS), an implementable version of cGLS, we utilize heteroskedastic inference to approximate the correlated errors.

Noting that data sets of observed decisions often may be small relative to the number of model parameters in practice, we employ bootstrapping to generate pseudo-data from which we learn additional estimators. The bootstrapping process allows us to derive an asymptotic approximation of the bias and standard error of an estimator. We utilize ensemble methods such as bagging, bumping, and gradient boosting to extract an estimator from the pseudo-data generated estimators that results in a reduced forecasting error. The ensemble methods are robust under noise and autocorrelated error terms.

To demonstrate the efficacy of the robust utility learning framework, we apply both to a toy example arising from Bertrand-Nash competition between two firms and to data from social game experiment designed to encourage energy efficient behavior among smart building occupants, which is described in depth in Chapter 4. The rest of the Chapter is organized as follows. We formulate in Section 5.1 the robust utility learning framework and provide an algorithm for implementing it. In Section 5.2 machine learning—ensemble learning techniques are presented for boosting estimation accuracy of the developed method. Section 5.3 contains the Bertrand-Nash competition example and in Section 5.4 we present the results of proposed utility learning methods applied to data from the social game. We make concluding remarks in Section 5.5.

5.1 Robust Utility Learning

In Chapters 2 and 3, we have explored utility learning and incentive design as a coupled problem in theory and in inference. We re-examine the utility learning problem using statistical methods that serve to improve estimation and prediction accuracy. Looking forward, our aim is to fold the new estimation scheme into the overall incentive design framework 2. This goal motivates why we are interested in learning more than a simple predictive model for agents, but rather a utility-based forecasting framework that accounts for individual preferences.

We parameterize agents utility function f_i by a general formulation using parameters $\theta_i = (\theta_{i1}, \ldots, \theta_{im_i}) \in \mathbb{R}^{m_i}$ and a finite set of basis functions $\{\phi_{ij}(x_i, x_{-i})\}_{j=1}^{m_i}$ such that

$$f_i(x; \theta_i) = \langle \phi_i(x_i, x_{-i}), \theta_i \rangle + \bar{f}_i(x) \tag{5.1}$$

where $\phi_i = [\phi_{i,1} \ \cdots \ \phi_{i,m_i}]^\top$ and $\bar{f}_i(x)$ is a function that captures *a priori* knowledge of the agent's utility function (e.g., the incentive component designed by the planner).

The basic utility estimation framework using equilibrium conditions for the game played between the players is well described in Chapter 3. In brief, assuming that each observation $x^{(k)}$ is an ε–approximate differential Nash equilibrium where the superscript notation $(\cdot)^{(k)}$

indicates the k–th observation. For each observation $x^{(k)}$, it may be the case that only a subset of the players, say $\mathcal{S}^k \subset \mathcal{I}$ at observation k, participate in the game. Let us now formulate a robust version of the utility learning framework that allows us to reduce our forecasting error and learn the noise structure which can be leveraged in extracting *pseudo–coalitions* between players which we describe in the sequel.

Define

$$X_i^{(k)} = \begin{bmatrix} D_i h_i(x_i^{(k)}) & D_i\phi_i(x^{(k)})) \\ \hat{h}_i(x_i^{(k)}) & \mathbf{0}_{\ell_i \times m_i} \end{bmatrix}, \tag{5.2}$$

where

$$\hat{h}_i(x_i) = \operatorname{diag}(h_{i,1}(x_i), \ldots, h_{i,\ell_i}(x_i)), \tag{5.3}$$

$$D_i h_i(x_i) = [D_i h_{i,1}(x_i) \ \cdots \ D_i h_{i,\ell_i}(x_i)], \tag{5.4}$$

and $n_d = (\ell_i + 1)n$ is the total number of data points. The regressor matrix is then defined as $X = \operatorname{diag}(X_1, \cdots, X_p) \in \mathbb{R}^{n_d \times (\ell_i+1)p}$ where $X_i = [(X_i^{(1)})^\top \ \cdots \ (X_i^{(n_i)})^\top]^\top$. Define the regression coefficient

$$\beta = [\mu_1^1 \ \ldots \ \mu_1^{\ell_1} \ \theta_1 \ \cdots \ \mu_p^1 \ \ldots \ \mu_p^{\ell_p} \ \theta_p]^\top \in \mathbb{R}^{(\ell_i+1)p} \tag{5.5}$$

and the observation matrix $Y = [Y_1 \ \cdots \ Y_p]^\top \in \mathbb{R}^{(\ell_i+1)p}$ where

$$Y_i = [\bar{f}_i(x^{(1)}) \ \mathbf{0}_{\ell_i} \ \cdots \ \bar{f}_i(x^{(n_i)}) \ \mathbf{0}_{\ell_i}]^\top. \tag{5.6}$$

Using the Euclidean norm for χ in (P) leads to an cOLS problem. Indeed,

$$\min_{\beta} \left\{ \|Y - X\beta\|_2 \big| \ \beta \in \mathcal{B} \right\} \tag{P1}$$

where $\mathcal{B} = \{\beta |\ \theta_i \in \Theta_i, \mu_i \geq 0, \ \forall i \in \mathcal{I}\}$. Enforcing that each of the constraint sets Θ_i is encoded by inequalities on θ_i, the above stated problem can be viewed as a classical multiple linear regression model with inequality constraints described by the data generation process

$$Y = X\beta + \epsilon, \ \beta \in \mathcal{B} \tag{5.7}$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_p)$ is the error term satisfying:

- $E(\epsilon|X) = 0^{n_d \times 1}$;
- $\operatorname{cov}(\epsilon|X) = \sigma^2 I^{n_d \times n_d}$;
- $\{\epsilon_i\}_{i=1}^p$ independent and identically distributed (iid) with a zero mean and σ^2 variance.

Error term that satisfies the above conditions lies in the assumptions of homoscedasticity. The core idea is that if the above assumptions hold, then the linear estimator satisfies the Markov BLUE property—best linear unbiased estimator. However, we assume ϵ is nonspherical [44]. With this general statistical model we are able to describe a data generation processes in which the error terms are correlated or lack constant variance. This fact will be leveraged in creating coalitions between players as we describe in Section 7.

Mathematically the nonspherical errors are modelled by

$$\operatorname{cov}(\epsilon|X) = G \succ 0,\ G \in \mathbb{R}^{n_d \times n_d}. \tag{5.8}$$

One drawback of this technique is that, given nonspherical standard errors, the cOLS estimator is biased—that is, it does not satisfy the Best Linear Unbiased Estimator (BLUE) property, a result of the Gauss–Markov theorem [44, Theorem 1, Chapter 5]. However, we can derive an unbiased estimator by multiplying (5.7) on the left with $G^{-\frac{1}{2}}$. This leads to the cGLS statistical model given by

$$(G^{-\frac{1}{2}}Y) = (G^{-\frac{1}{2}}X)\beta + (G^{-\frac{1}{2}}\epsilon),\ \beta \in \mathcal{B} \tag{5.9}$$

which now satisfies the BLUE property. In general, the explicit form of $\operatorname{cov}(\epsilon|X) = G$ is unknown. We use the residuals (5.7) to infer the noise by imposing structural constraints on G. We remark that there are many types of noise structures that can be used for imposing structure on G. We provide two example noise structures that could be used.

The first is a block diagonal structure [44, Chapter 5] in which $G = \operatorname{blkdiag}(K_1, \cdots, K_p) \in \mathbb{R}^{n_d \times n_d}$ where $K_i = \operatorname{blkdiag}(B_{i,1}, \ldots, B_{i,n_i}) \in \mathbb{R}^{(\ell_i+1)n_i \times (\ell_i+1)n_i}$ with each $B_{i,k} \in \mathbb{R}^{(\ell_i+1)\times(\ell_i+1)}$. To form estimates $\hat{K}_i$ of each K_i, we use the residuals of the cOLS estimate of β. Indeed, by writing $e = [e_1^\top \ \cdots \ e_p^\top]^\top$, we decompose the residual vector $e = Y - X\hat{\beta}_{\text{cOLS}} \in \mathbb{R}^{(\ell_i+1)n}$ into residuals for each player. Noting that there are n_i instances at which we have ℓ_i observations for player i, we let $(e_i)_{k,j} = (e_i)_{(\ell_i+1)(k-1)+j}$ where $k \in \{1, \ldots, n_i\}$ and $j \in \{1, \ldots, (\ell_i+1)\}$. Using the residuals, we compute $(\hat{B}_{i,k})_{j,j} = n_i^{-1}\sum_{t=1}^{n_i} e_{t,j}^2$ and $(\hat{B}_{i,k})_{l,j} = n_i^{-1}\sum_{t=1}^{n_i} e_{t,j}e_{t,l}$ for $j \neq l$ which are then used to populate the estimates $\hat{B}_{i,k} = [(\hat{B}_{i,k})_{l,j})]_{l,j=1}^{\ell_i+1} \in \mathbb{R}^{(\ell_i+1)\times(\ell_i+1)}$.

This particular noise structure is useful for problems in which the observations are multidimensional. Our formulation allows for multidimensional actions—although, for ease of presentation, we provide the details for scalar player decisions—and constraints which give rise to additional observations at each iteration k as can be seen in (5.2).

The second noise structure we consider is derived from the HC_4 estimator [33] and is given by

$$\hat{G} = \operatorname{diag}\left(\frac{e_1^2}{(1-b_1)^{\delta_1}}, \frac{e_2^2}{(1-b_2)^{\delta_2}}, \cdots, \frac{e_{n_d}^2}{(1-b_{n_d})^{\delta_{n_d}}}\right) \tag{5.10}$$

where $\delta_i = \min\{4, n_d b_i/(\sum_{i=1}^{n_d} b_i)\}$ and the b_i's are the diagonal elements of $B = X(X^\top X)^{-1}X^\top$. With this structure, the penalty for each residual increases with $b_i/\sum_{j=1}^{n_d} b_j$. As with the previous noise structure, we use the fitted cOLS estimator $\hat{\beta}_{\text{cOLS}}$ and residuals to get an initial $\hat{G}$. This noise structure is computationally efficient compared to many other noise structures.

In both cases, we substitute the inferred noise, $\hat{G}$, into the cGLS statistical model (5.9) to get the one–step constrained Feasible GLS (cFGLS) estimators. We iterate between the estimation of $\hat{G}$ and β_{cFGLS} either until convergence or for a fixed number of iterations to prevent overfitting. To resolve this trade-off and find the optimal iteration size we employ cross validation using Akaike information criterion (AIC) as score metric.

5.2 Boosting with Ensemble Methods

In this section, we describe several ensemble methods. Combined with a bootstrapping process, ensemble methods not only boost the size of what can often be a small data set in practice but also allow us to improve the estimator performance and explore the bias–variance tradeoff.

5.2.1 Bootstrapping and Bagging

Bootstrapping is a technique for asymptotic approximation of the bias and standard error [44, 46]. We employ *wild bootstrapping*—a technique consistent with heteroskedastic inference—to generate *pseudo–data* from which we generate many weak estimators that are combine using *bagging*—a technique that reduces overall variance [46].

In particular, we fit our cFGLS model which gives us $\hat{\beta}_{\text{cFGLS}}$. Then, generate N replicates of pseudo–data using the data generation process

$$\tilde{Y} = X\hat{\beta}_{\text{cFGLS}} + \Phi(e)\varepsilon, \tag{5.11}$$

where $\tilde{Y} \in \mathbb{R}^{n_d}$ is the new observation vector (pseudo-observations), $\hat{\beta}_{\text{cFGLS}} \in \mathbb{R}^{n_d}$ is the cFGLS estimator, $\varepsilon \sim N(0, I^{n_d \times n_d})$, $e \in \mathbb{R}^{n_d}$ is the residual vector given by $e = \tilde{Y} - X\hat{\beta}_{\text{cFGLS}}$ and $\Phi : \mathbb{R}^{n_d} \to \mathbb{R}^{n_d}$ is a nonlinear transformation such that $\Phi(e) = \hat{G}^{\frac{1}{2}} \in \mathbb{R}^{n_d \times n_d}$.

We use the data generation process in (5.11) to resample. These samples are drawn from iid random variables since $E(\Phi(e)\varepsilon|X) = \Phi(e)E(\varepsilon|X) = \Phi(e)E(\varepsilon) = \mathbf{0}_{n_d \times n_d}$.

Using the N replicates of pseudo–data generated by wild bootstrapping, we train N different models. We combine the resulting bootstrapped estimators by averaging:

$$\hat{\beta}_{\text{bag}} = \frac{1}{N}\sum_{j=1}^{N} \hat{\beta}_{\text{cFGLS},j} \tag{5.12}$$

where $\hat{\beta}_{\text{cFGLS},j}$ is the estimator using the j–th pseudo–data sample. Bagging works efficiently with high variance models and does not hurt the overall performance of the statistical model.

With this method, the empirical covariance matrix of $\hat{\beta}$,

$$\hat{C}_\beta = \frac{1}{N}\sum_{j=1}^{N} \left(\hat{\beta}_{\text{cFGLS},j} - \hat{\beta}_{\text{bag}}\right)\left(\hat{\beta}_{\text{cFGLS},j} - \hat{\beta}_{\text{bag}}\right)^\top, \tag{5.13}$$

is an asymptotic approximation of the covariance matrix and as such it reveals hidden structures between players that we leverage in the correlation utility learning procedure.

Algorithm 3 L_2–gradient boosting with cFGLS

function cFGLSgradboost(X,Y):
  $\hat{H} \leftarrow X(X^\top X)^{-1}X^\top$ ▷ compute $\hat{H}$ matrix
  $\nu \leftarrow s \in (0,1]$ ▷ set shrinkage (updating) parameter
  $m \leftarrow 1$, $M_{\max}$ ▷ stopping index and upper bound
  while $m < M_{\max}$ **do** ▷ Compute AIC's for each m
    $R_m \leftarrow (I_{n_d \times n_d} - \nu\hat{H})^m$
    $B_m \leftarrow (I_{n_d \times n_d} - R_m)$
    $\sigma_m^2 \leftarrow n_d^{-1}\sum_{i=1}^{n_d}(Y_i - (B_m Y)_i)^2$
    $\mathrm{AIC}(m) \leftarrow \left(\log\sigma_m^2 + \frac{1+(\mathrm{Tr}(B_m))/n_d}{1-(\mathrm{Tr}(B_m)+2)/n_d}\right)$
    AICs.append(AIC(m))
    $m \leftarrow m+1$
  $\hat{M} \leftarrow \arg\min_m \mathrm{AICs}$ ▷ find minimum point
  $\hat{\beta}_{\mathrm{cFGLS}} \leftarrow$ cFGLS estimate of β
  $e_{\mathrm{FGLS}} \leftarrow Y - X\hat{\beta}_{\mathrm{cFGLS}}$ ▷ residuals estimation
  $k \leftarrow 1$, $e \leftarrow e_{\mathrm{cFGLS}}$, $\hat{\beta}_{\mathrm{boost}} \leftarrow \hat{\beta}_{\mathrm{cFGLS}}$ ▷ initialize
  while $k < \hat{M}$ **do** ▷ Compute $\hat{\beta}_{\mathrm{boost}}$
    $\beta_i \leftarrow (X^\top X)^{-1}X^\top e$ ▷ residuals fitting
    $\hat{\beta}_{\mathrm{boost}} \leftarrow \hat{\beta}_{\mathrm{boost}} + \nu\beta_i$ ▷ update formula
    $e \leftarrow Y - X\hat{\beta}_{\mathrm{boost}}$ ▷ residuals update
    $k \leftarrow k+1$

5.2.2 Bootstrapping and Bumping

Here, we also use wild bootstrapping to generate pseudo–data, but in place of bagging, we use *bumping*—a technique for fitting cFGLS estimators using *stochastic search* over the model space [171].

We add the original training data sample to the N replicates of pseudo-data generated by the wild bootstrapping process and we use this data to learn $N+1$ cFGLS estimators. We evaluate these estimators on the training set and select the one with the least training error. Indeed, the bumping estimator is given by

$$\hat{\beta}_{\mathrm{bump}} = \arg\min_{\hat{\beta}_{\mathrm{cFGLS},j}} \|Y - X\hat{\beta}_{\mathrm{cFGLS},j}\|_2^2 \tag{5.14}$$

where $\hat{\beta}_{\mathrm{cFGLS},j}$'s are the cFGLS estimators from derived from the bootstrapped data, and Y and X are the training sets.

5.2.3 Gradient Boosting

Boosting methods are extremely useful for combining models where each new model is incrementally trained by emphasizing the errors of the previous training instances. Gradient

boosting is a technique that uses a loss function combined with a gradient (or sub-gradient) descent update method for combining weak learners at each iteration [45]. Several loss functions can be used like Huber or quadratic loss. We enabled an L_2–loss function for the gradient descent update method. This leads to a repeated residual fitting, which is applied until we reach iteration a stopping criteria selected using Akaike Information Criterion (AIC) to avoid overfitting [63] (early stopping). We combine gradient boosting with cFGLS and the procedure is detailed in Algorithm 3.

5.3 Bertrand-Nash Toy Example

Let us illustrate the framework and its performance of the robust utility learning framework before moving on by applying it to estimate market demand functions under Bertrand-Nash equilibrium (see, e.g., [12, 14]). The toy model can be thought of as an abstraction of Bertrand-price setting for commodities such as oil, gas, and coal [49].

Consider two firms competing to sell their product by setting the price p_1 and p_2 for firm 1 and 2, respectively. The firms utility functions are their revenue, i.e. $f_i(p_1, p_2) = p_i D_i(p_1, p_2, \xi)$ where D_i is the demand function for firm i and $\xi \sim \mathcal{N}(1.5, 0.5)$ is a random variable that captures the fact that demand is dependent on economic indicators in addition to the prices set by the firms. In this stylized example, we consider linear demand functions given by

$$D_i(p_1, p_2, \xi) = \theta_{i,1} + \theta_{i,2} p_1 + \theta_{i,3} p_2 + \nu\xi \tag{5.15}$$

where $\theta_i = (\theta_{i,j})_{j=1}^3$ are unknown parameters to be estimated and $\nu = 1.5$ is a known parameter. The prices are constrained to be in the interval $[0, \bar{p}]$ where $\bar{p} \in \mathbb{R}_+$ is the upper bound.

We let $\theta_1 = (-1.0, 0.5, -1)$ and $\theta_2 = (0.3, -1, 0.3)$ be the ground truth values for the parameters we wish to estimate. Thus, $\bar{f}_i(p_1, p_2) = \nu\xi$ and examining the marginal revenue functions $D_i f_i(p_1, p_2)$ we have that $\phi_1(p_1, p_2) = [1\ 2p_1\ p_2]^\top$, and $\phi_2 = [1\ p_1\ 2p_2]^\top$.

In order to generate the data set we add a noise term $\varepsilon \sim \mathcal{N}(0, 0.5)$ to the marginal revenue functions, i.e. $D_i f_i(p_1, p_2) + \varepsilon$, and solve for the Bertrand-Nash equilibrium. We simulate the game between the firms 600 times. In the robust utility learning framework, for this example, we employ the HC_4 noise structure and compute the cOLS, cFGLS, bagging, boosting and bumping estimators. We use a 10–fold cross validation proceedure to prevent over-fitting. Table 5.1 contains error using two metrics for both firms. Figure 5.1 shows the forecast for part of the testing set using cOLS and each of the ensemble methods as compared to the ground truth. While bagging performed best for firm 1 and boosting for firm 2 in the particular instantiation of this toy example, the performance more generally is dependent on the noise structure in the demand and marginal revenue functions, the sample size, and the dynamics between the two firms. However, it is interesting to point out that as we increase the variance on ξ, each of the ensemble methods performance stay relatively the same yet the cOLS error increases significantly.

Table 5.1: Mean Square Error (MSE) of forecasting using the proposed robust utility learning methods vs cOLS estimators for Bertrand-Nash competition. The best performing method is indicated in bold text for each of the firms.

Firm 1	**bagging**	boosting	bumping	cOLS
MSE	**0.05**	0.51	0.65	1.62
Firm 2	bagging	**boosting**	bumping	cOLS
MSE	1.58	**0.71**	0.89	2.54

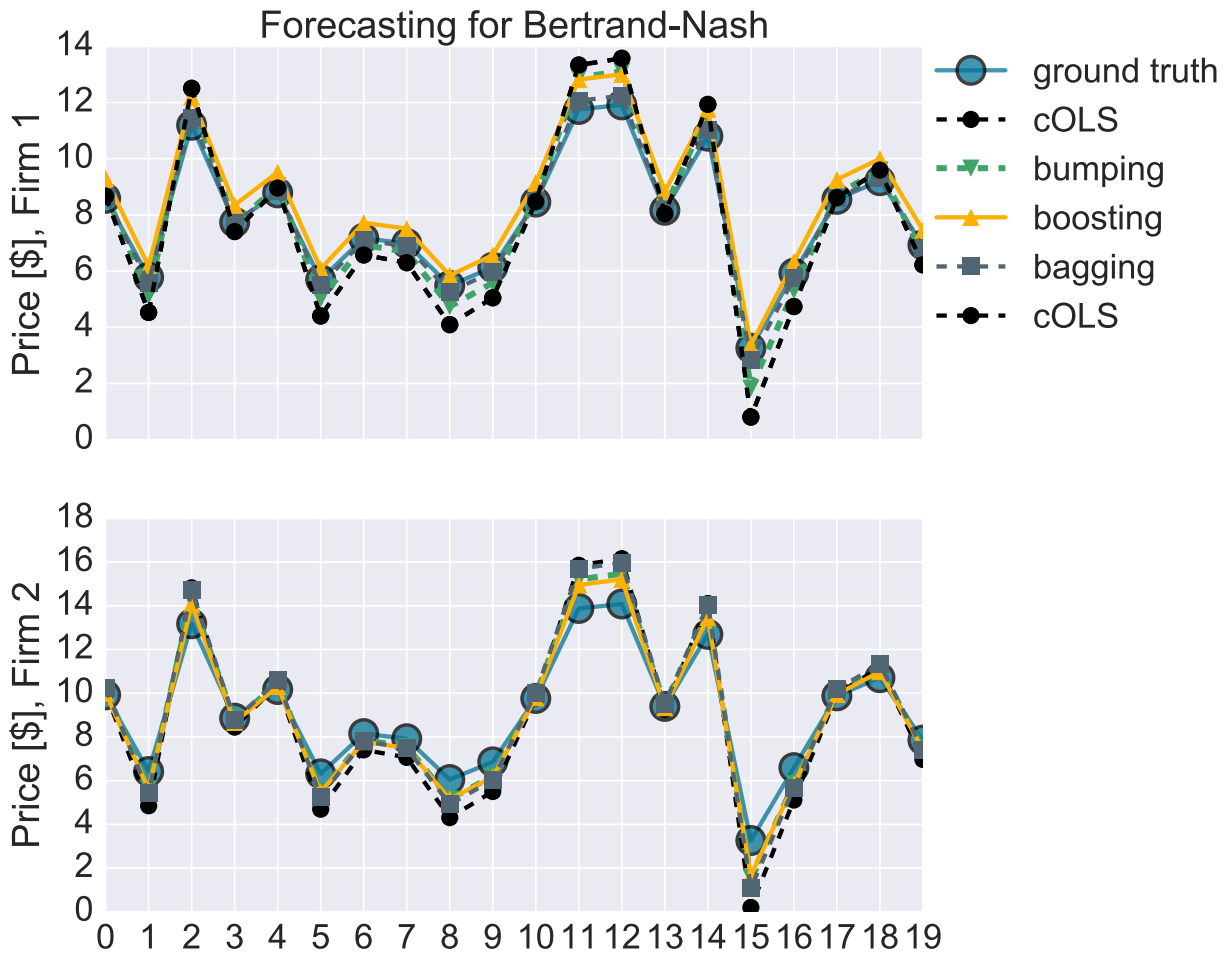


Figure 5.1: Forecast for Firms 1 & 2 using cOLS and each of the ensemble methods. The ground truth prices are depicted by the **blue dots**; the cOLS forecasts are depicted in **black**, the bagging forecasts are depicted in **gray**, the bumping forecasts are depicted in **green**, and the boosting forecasts are depicted in gold.

5.4 Forecasting via Robust Utility Learning

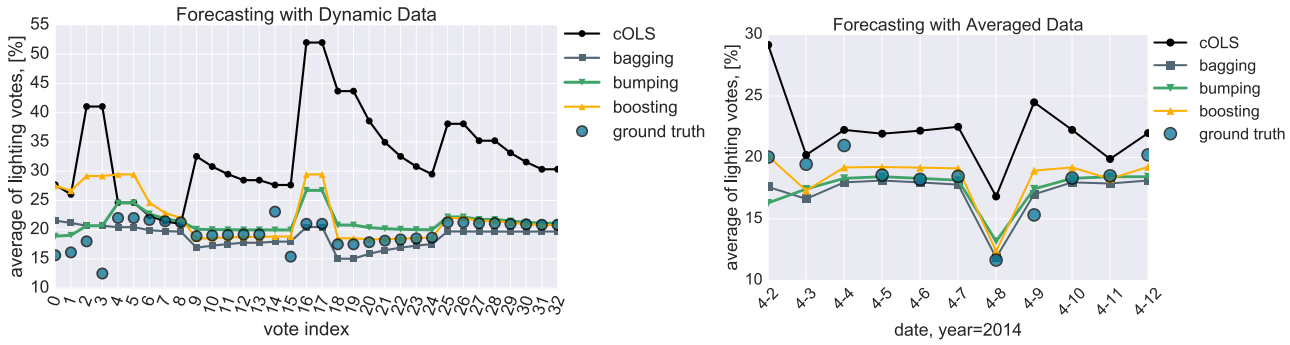
We apply the developed methods to the social game experiment described in Section 4.4 in order to to demonstrate their performance on a real data set. The social game experiment data set consists of the votes logged by the participants which vote throughout the day. We present estimation results for the complete data set of all the votes—which we refer to as the *dynamic data set*—and estimation results for an aggregated data set constructed by taking the average of a players' votes over the course of each day in the experiment—this is referred to as the *average data set*. While this aggregation significantly reduces the size of our data set, it smooths the players' voting profiles and increases the size of active players in each game—participants may arrive or leave the office when they so choose. The dynamic data set is much richer, being composed of every vote (a total of $6,885$ votes) the participants made over the course of the experiment (285 days). The time from one vote to the next may be several minutes to hours depending on the activity level of the participants. This data set allows us to extract more distinct player profiles and can support real-time incentive design schemes.

We present results for both data sets using data from the period of the experiment in which the default lighting setting was 20%—the results for the other default lighting settings are similar. The period of the experiment where the default lighting setting was 20% consisted of 42 days and thus the size of the averaged data set is 42. Over this period there were 220 votes by participants, which is the size of the dynamic data set. We divide each of the data sets into training (80% of the data) and testing (20% of the data) sets and apply each of the methods discussed in Section 5.1. We apply a 10–fold cross validation procedure to limit overfitting [46].

We estimate the parameters using cFGLS and the ensemble methods bagging, bumping, and boosting for both the average and dynamic data sets. For gradient boosting, we use the HC_4 noise structure since the values b_{ii} of B are large [33]; in each of the other methods, we used the block diagonal noise structure. Both noise structures are described in Section 5.1.

Using the estimated utility functions, we simulate the game using a projected gradient descent algorithm which is known to converge for concave games [43]. In Figure 5.2a and 5.2b, we compare the ground truth voting data to the predictions for each of the learning schemes using the dynamic and averaged data sets, respectively. Our proposed robust models—i.e. using the estimated parameters obtained via bagging, bumping, and boosting—capture most of the variation in the true votes (in both data sets) and significantly outperform cOLS.

In Table 6.1, using three metrics—Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Scaled Error (MASE)—we report the forecasting error for each of the methods. The estimated models using our robust utility learning methods significantly reduce the forecasting error when compared to cOLS. The cOLS method has particularly poor forecasting performance on the dynamic data set since it does not capture the correlated error terms describing the interactions between participants. Moreover, our robust methods perform better than cOLS with the averaged data set even though the sample size is small.



(a) (b)

Figure 5.2: Forecasting results for (a) dynamic data and (b) averaged data for the default lighting setting 20%. For the dynamic data, the x–axis values indicate the index of when a choice was made by one (or more) of the participants (the time from one index to the next may be several minutes to hours depending on the activity of the participants). For the averaged data, the x–axis values are dates (month and day). The y-axis values are the average of the votes where for each utility learning method we use the learned utilities to forecast the Nash equilibrium and the average of this voting profile across players is plotted. For comparison, we also provide the ground truth which is the average of the observed votes at each time instant. The forecast for the robust utility learning methods is approximately near the ground truth for both data sets while the cOLS estimates produce Nash equilibria with a large error.

As for the ensemble methods, bagging outperforms the other three methods when using the dynamic data set. For the averaged data set, gradient boosting gives the least forecasting error. This is in large part due to the fact that we use the HC_4 noise structure. Since the average data set has been smoothed, we expect less correlation between players and the HC_4 noise structure captures this.

5.4.1 Bias Approximation and Bias–Variance Tradeoff

Forecasting accuracy can be enhanced by allowing for a small amount of bias if it results in a large reduction in variance. For a process $Y = X\theta + \epsilon$, the Mean Square Error (MSE) characterizes the *bias–variance tradeoff*:

$$\begin{aligned} \text{MSE}(x) &= E[(Y - \theta_{\text{est}}^\top x)^2] && (5.16)\\ &= \underbrace{(E[\theta_{\text{est}}^\top x] - Y)^2}_{\text{bias}} + \underbrace{E[(\theta_{\text{est}}^\top x - E[\theta_{\text{est}}^\top x])^2]}_{\text{variance}} && (5.17) \end{aligned}$$

In our robust utility learning framework, we introduce noise structures that approximate the true data process in order to fit cFGLS estimators that are nearly unbiased for those players whose historical voting record has a large amount of variation.

Table 5.2: Forecast errors as measured by Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Scaled Error (MASE) [64] using the proposed robust utility learning methods for both data sets in default lighting setting 20. Forecast errors are computed by comparing the average of the ground truth votes to the average of the forecasted Nash equilibrium. The best performing method is indicated in bold text for each of the data sets, dynamic and average.

Dynamic, $\hat{f}_i$	**bagging**	boosting	bumping	cOLS
RMSE	**8.31**	10.11	12.56	22.53
MAE	**5.20**	6.55	6.38	18.35
MASE	**2.08**	6.38	2.55	7.34
Averaged, $\hat{f}_i$	bagging	**boosting**	bumping	cOLS
RMSE	2.05	**1.68**	1.96	9.36
MAE	1.58	**1.31**	1.48	6.01
MASE	0.71	**0.59**	0.67	2.69

Table 5.3: The cFGLS estimator value and the bagging, gradient boosting and bumping ensemble methods bias approximation for the most active participants. We utilized the dynamic data set from the period in which the default lighting setting was set to 20. In bold, we denote the players with nearly unbiased estimators.

Id	cFGLS	Bagging Bias	Boosting Bias	Bumping Bias
2	**-0.7**	**0.11**	**0.17**	**0.02**
6	0.5	1.12	1.77	0.93
8	298.1	-176.9	-370.3	120.5
14	337.5	-186.3	-400.2	149.7
20	**-0.8**	**0.07**	**0.21**	**-0.53**

We approximate the bias for each of the estimators. In Table 5.3, we present cFGLS estimates obtained using the dynamic data during the time window in which the default lighting setting was 20%[1] for selected players—the most active players—as well as the approximated bias for the estimates generated by bagging, bumping, and boosting.

Figures 5.3 and 5.4 contain histograms of the cFGLS estimators obtained using the bootstrapped average and dynamic data, respectively. In each of these histograms, we indicate the original cFGLS [2] (indicated in **red**), bagging (indicated in **blue**), bumping (indicated in **green**), and boosting (indicated in **orange**) estimators with dashed vertical lines.

The histogram in Figure 5.3 contains the cFGLS estimators for player 2. This histogram is representative of the other players for the average data set. We see that the original

[1]The results for the other default lighting settings are similar.

[2]This is the cFGLS estimator produced using the original average and dynamic data sets and not the bootstrapped data sets.

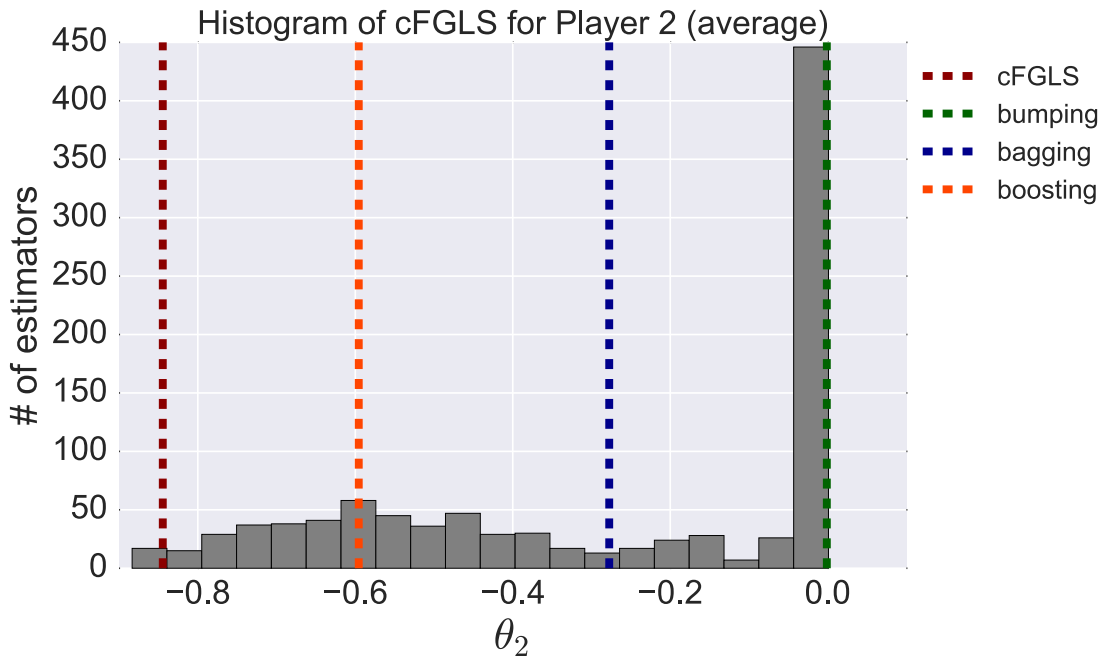
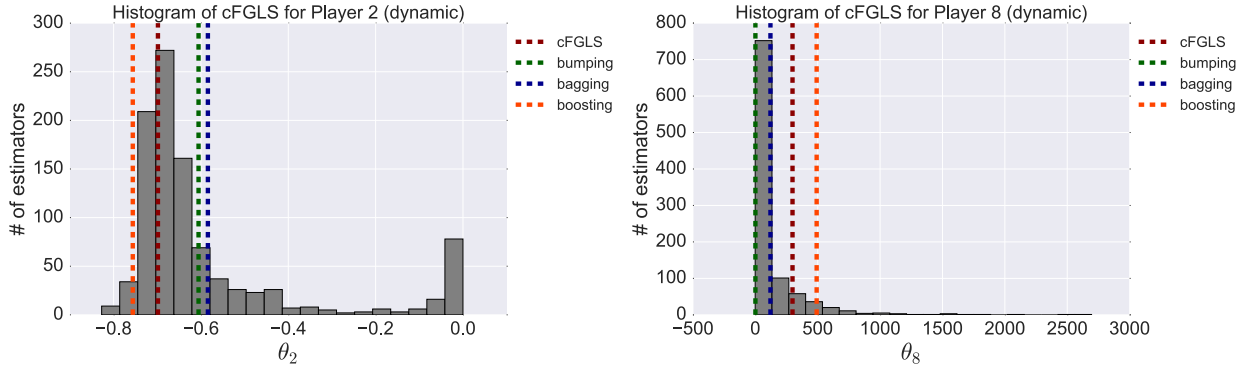


Figure 5.3: The histogram depicts estimator values for player 2 using the wild bootstrapping technique using the *average data set*. The vertical lines mark the value of the **cFGLS (red)**, **bumping (green)**, **bagging (blue)**, and **boosting (orange)** estimators. We remark that the estimators are all biased. This is expected due to limited sample size of the average data set. Thus, the average data set cannot be used for optimizing the bias-variance tradeoff.

cFGLS, bagging, bumping, and boosting estimators each show some amount of bias. This is largely due to the fact that the average data set has a small sample size.

In Figure 5.4 we show the histogram of cFGLS estimators for players 2 and player 8 produced via bootstrapped dynamic data. We can see that the original cFGLS estimator (vertical **red** line) is nearly unbiased for player 2 (refer to 5.4a), indicated by the approximate Gaussian distribution around the cFGLS estimate. This is generally true for the participants with the most variation and frequency in their voting record. Player 2 is one such participant. However, bagging, bumping, and boosting produce estimates which are slightly biased in exchange for a reduction in estimator variance—see (5.16).

While a very active voter, frequently participating in the game, player 8's voting record has little variation (the majority of the time it votes for 0% dim level). Figure 5.4b contains the cFGLS estimators for player 8 and we see that each of the estimators are slightly biased.



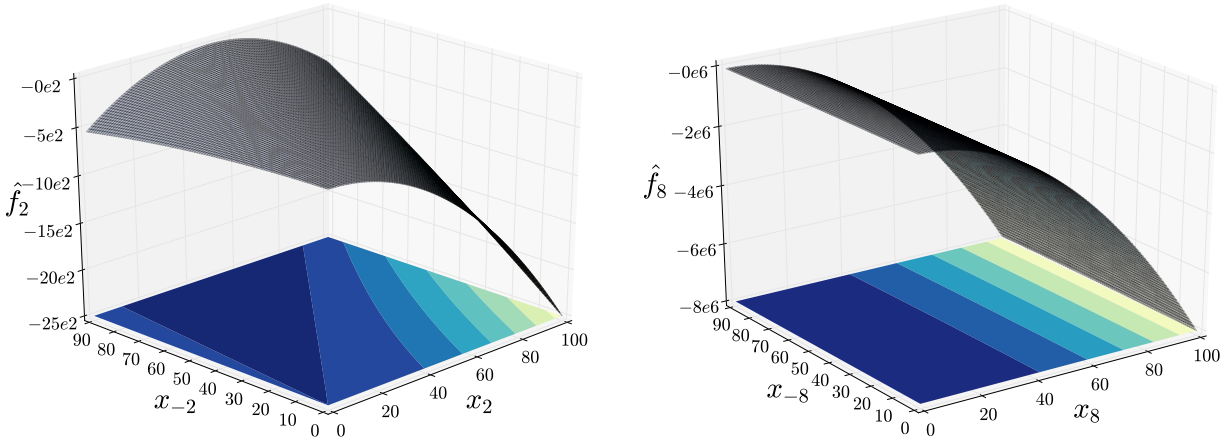
(a) (b)

Figure 5.4: The histograms depict the estimates generated with the wild bootstrapping technique using the *dynamic data set* for (a) player 2 and (b) player 8. The vertical lines mark the value of the **cFGLS (red)**, **bumping (green)**, **bagging (blue)**, and **boosting (orange)** estimators. The histogram for player 2 is approximately normally distributed around the initial cFGLS estimator, indicating that it is unbiased. Yet, this is not the case for player 8 and thus, its cFGLS estimator is biased. Overall, the majority of the proposed ensemble methods result in a significant reduction in variance in exchange for an small increase in bias and greater forecasting accuracy. In our other work [100], we develop a hierarchical mixture model that considers both bias and variance.

5.4.2 Estimated Utility Functions

Figure 5.5 shows the estimated utility functions and their contour plots for players 2 and 8—passive and aggressive players, respectively—using the parameters obtained via the bagging ensemble method with the dynamic data set. We remark that we do not observe the actual value of the participants' utilities; we instead observe only their decisions. The purpose of the figures is to show the estimated utility shapes for players with significantly different voting profiles (the observable we have). The particular players selected represent participants that prefer *winning* to lighting satisfaction (e.g., player 8) and participants that prefer *lighting satisfaction* to winning (e.g., player 2). In particular, player 2's estimated utility function appears to be higher at greater lighting settings. Exactly the opposite occurs for player 8 whose estimated utility function indicates that despite changes in the average lighting vote of other players, player 8 aggressively votes for a zero lighting setting which returns the most points.

For comparison—and to highlight the improvement that the robust utility learning framework offers—in Figure 5.6 we show the estimated utility function for player 8 using cOLS. What we see is a very different utility function that indicates player 8 cares more about lighting satisfaction than winning—indicated by the fact that its utility is not maximized at zero. This is misleading since player 8 predominately votes for zero and it is significant since



(a) (b)

Figure 5.5: Bagging estimated utility functions—using the dynamic data set—of (a) player 2 and (b) player 8. The functions are plotted as a function of each player's own vote x_2 (resp. x_8) and other players' votes x_{-2} (resp. x_{-8}). Notice that player 8, an aggressive player, is indifferent to the choices of the other participants as indicated by the fact that its utility is maximized in the same location given any value of x_{-8}. On the other hand, player 2 responds to changes in the other participants' votes and appears to prefer a greater lighting settings (more illumination). This indicates that there are different types of players and thus, incentives may need to be designed individually for these player types in order to elicit the desired response.

incentive/control design based on such an erroneous utility function may lead to very poor performance and participant dissatisfaction.

5.5 Chapter Summary

In this Chapter we introduced a general framework for robust utility learning using a heteroskedastic inference adaptation to cGLS and we extended our framework by leveraging ensemble machine learning. To demonstrate the utility learning methods, we applied them to data collected from smart building social game (in section 4.4) we conducted where occupants vote for shared resources and participate in a lottery. Part of our results is that we are able to estimate nearly unbiased estimators for several agent profiles and significantly reduce the forecasting error as compared to cOLS. Our robust framework is modular in that it can be extended to other choices of utility functions that incorporate different basis functions and easily generalized to other application domains.

Table 5.4: Estimated covariance matrix for the most active players using the (a) dynamic data set and (b) average data set. The colored column-row pairs indicate the players whose utilities we modify to generate the correlated game; the column indicates the player(s) whose estimated parameter is used to modify the row player's utility. Player 2 and 14 are anti-correlated and player 8 and 14 (respectively, 2 and 20) are positively correlated. Players 2 and 20 are passive, voting more for lighting satisfaction than winning, where players 8 and 14 vote more aggressively.

(a) Average Data

Id	**2**	6	**8**	**14**	**20**
2	0.086	0.080	-0.190	-0.248	**0.059**
6	0.080	7.56	8.64	9.02	0.028
8	-0.190	8.64	170.98	**44.29**	-0.337
14	**-0.248**	9.02	**44.29**	87.34	-0.312
20	0.059	0.028	-0.337	-0.312	0.063

(b) Dynamic Data

Id	**2**	6	**8**	**14**	**20**
2	0.044	0.059	-2.805	-5.191	**0.031**
6	0.059	7.836	-16.82	0.844	-0.016
8	-2.805	-16.82	6.43×10^4	$\mathbf{4.28\times10^4}$	-7.60
14	**-5.191**	0.844	$\mathbf{4.28\times10^4}$	8.84×10^4	-12.59
20	0.031	-0.016	-7.60	-12.59	0.073

The robust utility learning framework enables us to effectively *close the loop* around smart building occupants by providing the foundation for learning a decision-making model that can be integrated into the incentive or control design process. While we apply the method to smart building social game data, it can be applied more generally to scenarios with the task of inverse modeling of competitive agents and provides a useful tool for many smart infrastructure applications where learning decision–making behavior is crucial.

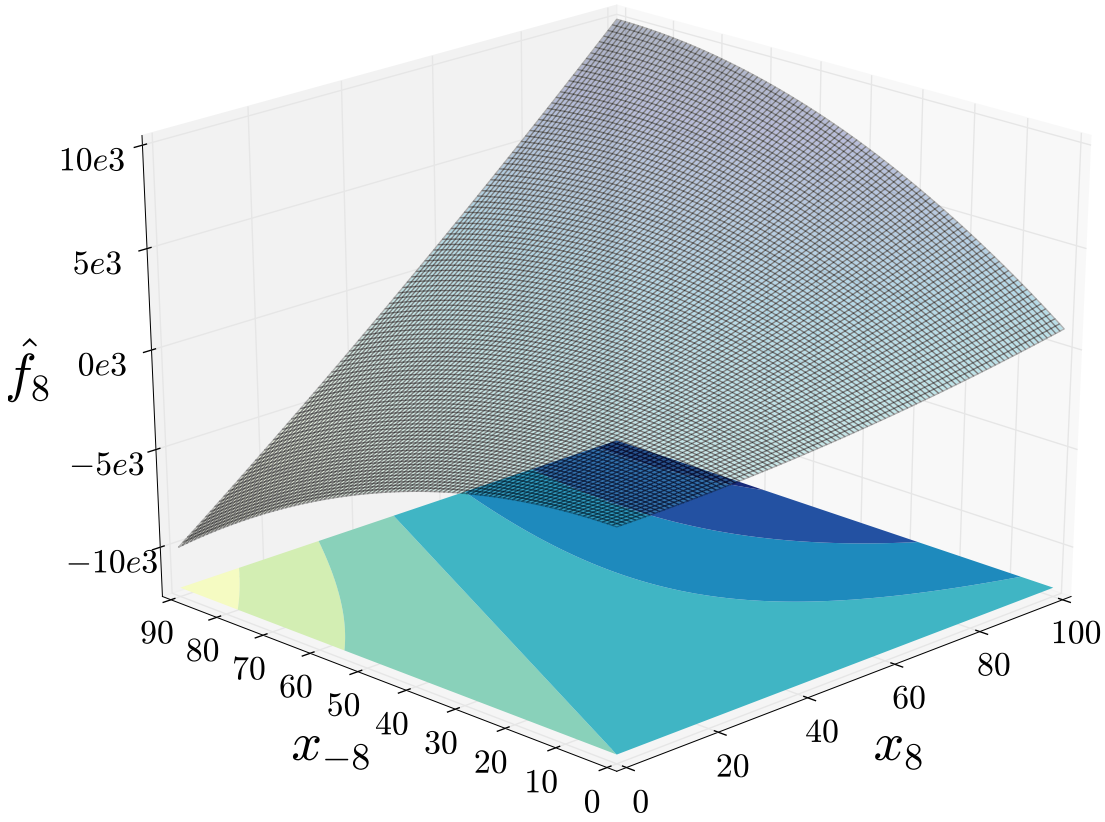


Figure 5.6: Player 8's cOLS estimated utility function—using the dynamic data set—plotted as a function of (x_8, x_{-8}). This figure demonstrates that using cOLS (the worst performing estimator) results in learning a utility function that is not representative of this type of player's behavior (as can be seen by comparing to Figure 5.5b). Incentives or control designed using this function may result in poor performance.

Chapter 6

Utility Learning via Mixture of Utilities

Utility learning problem is challenging, especially taking in account correlated noise from agents interaction. In its base formulation—i.e. Chapter 3, forecasting error can be high and the learning problem is vulnerable in attacks (see Chapter 4). However, there are some important questions in utility learning that worth answering:

Can we represent utility functions in a hierarchical probabilistic framework? Can we model an agent's utility as a combination of several utilities? Then, how can we learn this dynamic probabilistic utility model? Does the probabilistic utility learning framework model temporal dependencies?

In the current Chapter, we develop the theoretical formulation of a new parametric utility learning method that uses a probabilistic interpretation—i.e. a mixture of utilities—of agent utility functions. The mixture of utilities modeling paradigm allows us to account for variations in agents' parameters over time. Our method combines resulting utility learners under non-spherical noise terms. The main contribution is the adaptation to utility learning model of a probabilistic framework leveraging Hierarchical mixtures of experts using an EM algorithm for the inference [87]. Several levels of softmax functions used as gates and at the lowest level of the proposed graphical model softmax functions, experts nodes, are used for combining regression models. The resulting scheme is a *Mixture of constrained Feasible Generalized Least Squares* (Mix-cFGLS) which uses heteroskedasticity inference for correlated errors in the resulting regression model. Mix-cFGLS is a statistical model that we show is a powerful tool for utility learning that providing greater accuracy in the prediction of agents' responses. Furthermore, captured in this framework is the fact that agents' utility functions are not static; instead, the parameters of players utility functions can depend on historical data.

In this Mix-cFGLS framework we explore the trade-off between minimizing bias and minimizing the variance of predictions. Using Mix-cFGLS for utility learning, it allows for a

small amount of bias results in a substantial decrease in variance and increase in forecasting accuracy. Moreover, the benefit of using a mixture of utilities framework is that it allows us to capture the effects of different environmental conditions on the outcome of decisions. For example, perhaps one utility function better models a person in the morning and another in the evening; through mixing multiple utilities, we can capture this behavior in a single model. This is an important generalization of classical game theoretic models in which utility functions are static [129].

The performance of the proposed method is shown by estimating the utility functions of players using data from social game experiment presented in section 4.4. Using occupant voting data we simulate the new game defined by the estimated mixture of utilities and show that the resulting forecast is more accurate than robust utility learning methods such as constrained Feasible Generalized Least Squares (cFGLS), ensemble methods such as bagging, and classical methods such as Ordinary Least Squares (OLS). Clearly, a graphical representation of utility learning framework has impressive potentials for forecasting agents' actions.

6.1 Hierarchical Mixture of Experts: A Probabilistic Framework for Utility Learning

Motivated by the utility learning framework under a non–spherical noise assumption introduced in 5, we extend it and develop the core theoretical formulation of the utility learning framework for a mixture of utilities. Proposed mixture of utilities can be applied in both non-cooperative games and discrete choice models described in the generalized gamification framework in Chapter 2.

6.1.1 Utility Estimation—Mixture of cFGLS

We extend the robust learning framework described in Chapter 5 to a probabilistic setting that allows us to learn a mixture of utilities. As described above, this adaptation gives us a more complex representation of the decision-making model for agents. Strengthen by the cFGLS learners under non-spherical noise inference we adapt a Hierarchical mixture of experts with regression emissions—models under Heteroskedasticity and introduce a Mixture of Constrained Feasible Generalized Least Squares (Mix-cFGLS) framework. Figure 6.1 shows a type of tree structure probabilistic model (using two-levels) with several gating layers giving a soft probabilistic decisions splits. Each split is a probabilistic function of a linear combination of inputs and the hierarchical structure lead to terminal—leaf nodes, which are the "experts". Expert nodes in the utility learning probabilistic adaptation model is the interaction of the response variable and the input P1. The top gating sigmoid functions

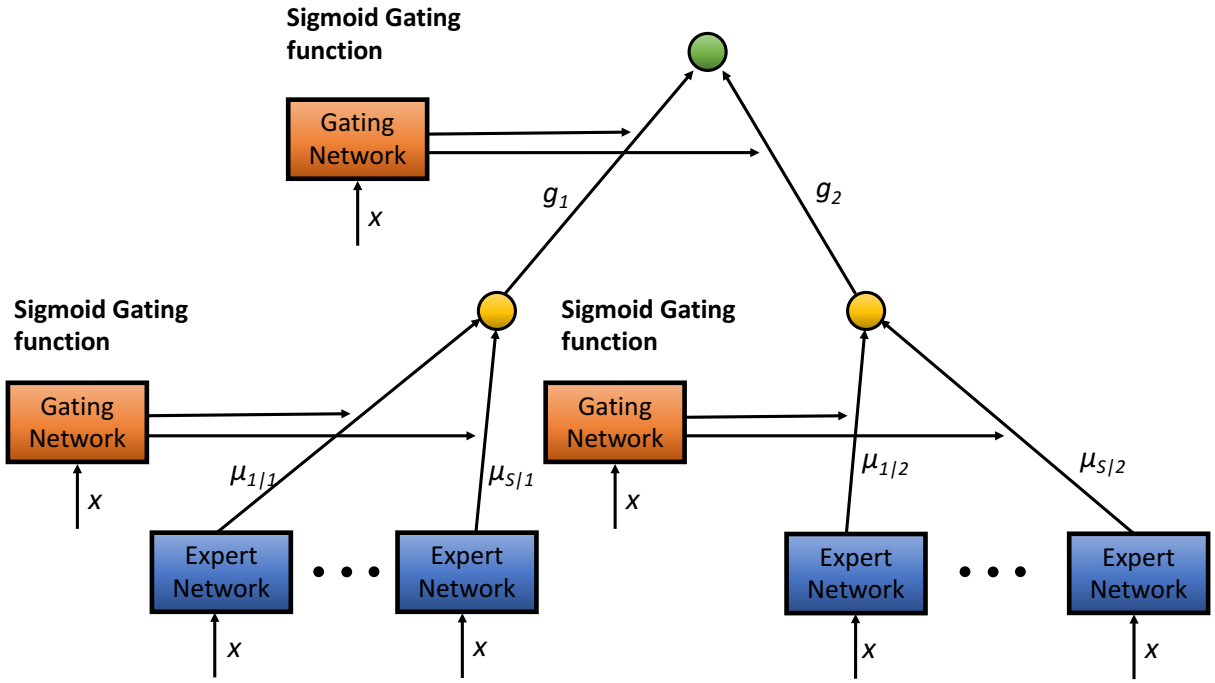


Figure 6.1: A two-level hierarchical mixture of experts probabilistic graphical model representing the utility function—learning of an agent.

in Figure 6.1 are defined as

$$g_i(\mathrm{x}, \lambda_i) = \frac{\exp\left(-(\lambda_i)^\top \mathrm{x}\right)}{\sum_{i=1}^{M} \exp\left(-(\lambda_i)^\top \mathrm{x}\right)}, i = 1, 2, \cdots, M \tag{6.1}$$

for a M-way split. Then in the next level the gating sigmoid functions are as follows

$$\mu_{j|i}(\mathrm{x}, \phi_j^i) = \frac{\exp\left(-(\phi_j^i)^\top \mathrm{x}\right)}{\sum_{j=1}^{K} \exp\left(-(\phi_j^i)^\top \mathrm{x}\right)}, j = 1, 2, \cdots, K \text{ and } i = 1, 2, \cdots, M \tag{6.2}$$

The Hierarchical probabilistic model assigns probability to j-th branch of the graphical model in Figure 6.1 given assigned probability to the i-th branch at the level above. Next, each expert—leaf node has the utility learning probabilistic adaptation model is the interaction of the response variable and the input P1. Then, the probabilistic model for player is given by

$$p(Y_i|X_i, \beta, \phi_j^i, \lambda_i) = \sum_{j=1}^{K} \mu_{j|i}(\mathrm{x}_s, \phi_j^i) \sum_{i=1}^{M} g_i(\mathrm{x}, \lambda_i) \cdot N(Y|(\beta j|i)^\top \mathrm{x}, G \succ 0, \beta j|i > \beta_{LB})) \tag{6.3}$$

where $\beta_{LB} = [0\ 0\ \theta_{LB}]^\top \in \mathbb{R}^3$,

$$Y = [-D_i\psi_i(x^{(1)})\ 0\ 0\ \cdots\ -D_i\psi_i(x^{(n_i)})\ 0\ 0]^\top$$

is an observation–dependent vector, x_i is a covariate vector and is a transposed row of the regressor–design matrix $X_i = [(X_i^{(1)})^\top \ \cdots \ (X_i^{(n_i)})^\top]^\top$, and $G \succ 0$ is the non-spherical structure that models the noise term of the statistical process of player i's utility learning problem. For the estimation, we maximize the log-likelihood over the whole data-set.

A mixture of regression models is a statistical model that splits data points into several subareas to which a regression learner is fit. Unfortunately, in real applications we do not have *a priori* knowledge of each subarea's data structure and density. Thus, the Mix-cFGLS procedure has to learn, in parallel, both the subareas' data point density plus the best regression learner that fits the data points in these subareas. This is achievable by the Expectation-Maximization (EM) algorithm over the cost function resulting from the complete likelihood of our statistical model. EM algorithm is used to optimize a non-convex cost—objective function resulting from the log-likelihood over the whole given data-set. EM algorithm can be approximated using sampling techniques [51, 180] and also can run using on-line learning formulations [87].

To formalize the proposed extension, we first describe the form of the mixture model. Let x_i^l be the l–th row of X_i and y_i^l be the l–th row of Y_i. Note that each of the rows of X_i depends on one of the data points $x^{(k)}$ where k indexes the data point. Without loss of generality we use a Hierarchical mixture model with one level. Then this specific mixture model is given by

$$\theta_i^{\text{mix}}(x) = \sum_{s=1}^{M} \pi_i^s(\mathrm{x}_i^l, \xi_i^s)\theta_i^s \tag{6.4}$$

where $\pi_i^s(\mathrm{x}_i^l, \xi_i^s)$ is a softmax function defined by

$$\pi_i^s(\mathrm{x}_i^l, \xi_i^s) = \frac{\exp\left(-(\xi_i^s)^\top \mathrm{x}_i^l\right)}{\sum_{s=1}^{M} \exp\left(-(\xi_i^s)^\top \mathrm{x}_i^l\right)} \tag{6.5}$$

and the ξ_i^s's are the different vectors that govern the softmax function. In particular, the softmax function acts as a gate for calculating the distribution probability of each θ_i^s. The ξ_i^s's are also unknown to building manager and must be learned along with the parameters θ_i^s. The softmax function assigns a probability to each θ_i^s based indirectly on past actions through x_i^l and it is a continuous probability density and sums to one—i.e. $\sum_{s=1}^{M} \pi_i^s(\mathrm{x}_i^l, \xi_i^s) = 1$.

6.1.2 Expectation-Maximization Algorithm & Inference

Let us introduce some additional notation. Define $\xi_i = (\xi_i^1, \ldots, \xi_i^M)$, $\beta_i = [\beta_i^1 \ \cdots \ \beta_i^M] \in \mathbb{R}^{3M}$ with $\beta_i^s = (\theta_i^s, \mu_i^1, \mu_i^2)$ being the regression coefficient for mixture component s, and $\xi_i = [\xi_i^1 \ \cdots \ \xi_i^M] \in \mathbb{R}^{3M}$ with ξ_i^s being the coefficient of the softmax for mixture component s. Recall that $\pi_i^s(\mathrm{x}_i, \xi_i^s)$ is the mixture coefficient probability distribution governed by a softmax function (see (6.5)). The probabilistic model for player i is given by

$$p(Y_i|X_i, \beta_i, \xi_i) = \sum_{s=1}^{M} \left(\pi_i^s(\mathrm{x}_i, \xi_i^s) \cdot N(Y_i|(\beta_i^s)^\top \mathrm{x}_i, G_i \succ 0, \beta_i^s > \beta_{i,LB}^s)\right) \tag{6.6}$$

where $\beta^s_{i,LB} = [0\ 0\ \theta_{LB}]^\top \in \mathbb{R}^3$,

$$Y_i = [-D_i\psi_i(x^{(1)})\ 0\ 0\ \cdots\ -D_i\psi_i(x^{(n_i)})\ 0\ 0]^\top$$

is an observation–dependent vector, x_i is a covariate vector and is a transposed row of the regressor–design matrix $X_i = [(X_i^{(1)})^\top\ \cdots\ (X_i^{(n_i)})^\top]^\top$, and $G_i \succ 0$ is the non-spherical structure that models the noise term of the statistical process of player i's utility learning problem.

Given a data set for player i and assuming i.i.d. observations $\mathcal{O}_i = \{(\mathrm{x}_i^l, \mathrm{y}_i^l) : l = 1, \cdots, 3n_i\}$, where $\mathrm{x}_i^l, \mathrm{y}_i^l$ are the l-th rows of X_i, Y_i respectively, the log-likelihood function of the model is given by

$$L(\beta_i, \xi_i|\mathcal{O}_i) = \sum_{l=1}^{3n_i} \log\left[\sum_{s=1}^M \pi_i^s(\mathrm{x}_i^l, \xi_i^s) \cdot N(Y_i|(\beta_i^s)^\top \mathrm{x}_i^l, G_i \succ 0, \beta_i^s > \beta_{i,LB}^s)\right] \tag{6.7}$$

Our goal is to optimize the cost function (6.7) using the EM algorithm. In support of this, we introduce a set of binary latent variables $Z_i = \{z_i^l\}$ where $z_i^l = (z_i^{l,1}, \ldots, z_i^{l,s})$ and such that $z_i^{l,s} \in \{0, 1\}$. That is, these latent variables serve to split the set of points $(\mathrm{x}_i^l, \mathrm{y}_i^l)$ for $l \in \{1, \ldots, 3n_i\}$ into subsets, one for each mixture model $s \in \{1, \ldots, M\}$.

Unlike existing models in the literature, for our particular case, it is the case that the data point $x^{(k)}$ for each k generates three rows of X_i,

$$X_i^{(k)} = \begin{bmatrix} D_i h_{i,1}(x_i^{(k)}) & D_i h_{i,2}(x_i^{(k)}) & D_i\phi_i(x^{(k)})) \\ h_{i,1}(x_i^{(k)}) & 0 & 0 \\ 0 & h_{i,2}(x_i^{(k)}) & 0 \end{bmatrix}, \tag{6.8}$$

and three rows of Y_i,

$$Y_i^{(k)} = [-D_i\psi_i(x^{(k)})\ 0\ 0]^\top. \tag{6.9}$$

To ensure that the rows x_i^l and y_i^l of X_i and Y_i respectively that are generated by a data point $x^{(k)}$ are assigned to the same mixture model s, we use a soft-thresholding [108] heuristic in which we average the three $\tau_i^{l,s}$ corresponding to the rows x_i^l and y_i^l generated by a single data point $x^{(k)}$. We use this average to assign the three rows of X_i and Y_i corresponding to a particular $x^{(k)}$ to mixture models with the weight of assignment being the average of the three $\tau_i^{l,s}$'s, one for each row. Note the averaging process may result in the average of the expected value of the $z_i^{l,s}$ being in the interval $[0, 1]$ instead of in the set $\{0, 1\}$. This is why the assignment is weighted according to the average. An alternative heuristic would be to use hard-thresholding [27] in which the assignment of the three rows to a particular model is selected based on thresholding the average.

Combining $Z_i = \{z_i^l\}$ with $\mathcal{O}_i$, we have a new data set $\mathcal{O}_i^C = \{(\mathrm{x}_i^l, \mathrm{y}_i^l, z_i^l) : l = 1, \cdots, 3n_i\}$.

The *complete* log-likelihood for (6.6) is given by

$$L_C(\beta_i, \xi_i|\mathcal{O}_i^C) = \sum_{l=1}^{3n_i}\sum_{s=1}^M z_i^{l,s} \log\left[\pi_i^s(\mathrm{x}_i^l, \xi_i^s) \cdot N(\mathrm{y}_i^l|(\beta_i^s)^\top \mathrm{x}_i^l, G_i \succ 0, \beta_i^s > \beta_{i,LB}^s)\right] \tag{6.10}$$

However, since we do not observe the latent variables Z_i, we compute their posterior probability in the expectation step (*E–step*) of the EM algorithm. The posterior probability of latent variables Z_i is given by

$$\tau_i^{l,s} = E[z_i^{l,s}] = p(z_i^{l,s} = 1|X_i, Y_i) = \frac{\pi_i^s(\mathrm{x}_i^l, \xi_i^s) N(\mathrm{y}_i^l|(\beta_i^s)^\top \mathrm{x}_i^l, G_i \succ 0, \beta_i^s > \beta_{i,LB}^s)}{\sum_{\ell=1}^M \pi_i^\ell(\mathrm{x}_i^l, \xi_i^\ell) N(\mathrm{y}_i^l|(\beta_i^\ell)^\top \mathrm{x}_i^l, G_i \succ 0, \beta_i^\ell > \beta_{i,LB}^\ell)}. \quad (6.11)$$

Next in the maximization step (*M–step*) of the EM algorithm, we maximize the following objective function:

$$L_C(\beta_i, \xi_i|\mathcal{O}_i^C) = \sum_{l=1}^{3n_i} \sum_{s=1}^M \tau_i^{l,s} \log \left[\pi_i^s(\mathrm{x}_i^l, \xi_i^s) \cdot N(\mathrm{y}_i^l|(\beta_i^s)^\top \mathrm{x}_i^l, G_i \succ 0, \beta_i^s > \beta_{i,LB}^s)\right]$$
$$= \sum_{l=1}^{3n_i} \sum_{s=1}^M \tau_i^{l,s} \log \left[N(\mathrm{y}_i^l|(\beta_i^s)^\top \mathrm{x}_i^l, G_i \succ 0, \beta_i^s > \beta_{i,LB}^s)\right] + \sum_{l=1}^{3n_i} \sum_{s=1}^M \tau_i^{l,s} \log \left[\pi_i^s(\mathrm{x}_i^l, \xi_i^s)\right] = Q_{\beta_i, \xi_i}.$$

The solution of the optimization problem at the *M–step* is given by solving the following completely decoupled optimization problems. First, we minimize the cost function

$$\hat{Q}_{\beta_i} = \sum_{l=1}^{n_i} \sum_{s=1}^M \tau_i^{l,s} \log \left[N(\mathrm{y}_i^l|(\beta_i^s)^\top \mathrm{x}_i^l, G_i \succ 0, \beta_i^s > \beta_{i,LB}^s\right] \quad (6.12)$$

with respect to $\beta_i = (\beta_i^1, \ldots, \beta_i^M)$. The above cost function is solved using a weighted constrained Iteratively Reweighted Least Squares (IRLS) algorithm using data pairs $\{\mathrm{x}_i^l, \mathrm{y}_i^l\}$ with weights $\tau_i^{l,s}$. Then, we minimize the cost function

$$\hat{Q}_{\xi_i} = \sum_{l=1}^{3n_i} \sum_{s=1}^M \tau_i^{l,s} \log \left[\pi_i^s(\mathrm{x}_i^l, \xi_i^s)\right] \quad (6.13)$$

with respect to $\xi_i = (\xi_i^1, \ldots, \xi_i^M)$. The above cost function is optimized using an IRLS algorithm with data pairs $\{\mathrm{x}_i^l, \tau_i^{l,s}\}$.

In summary, we propose Algorithm 4 for solving EM for the utility learning problem cast as a Mix-cFGLS problem. A key aspect to the solution is in selecting an appropriate noise structure for each player's data structure. We initially fit the data using cFGLS, initialize our algorithm and use the estimated $\hat{G}$ in the EM update steps. Since EM is a coordinate descent algorithm for the non-convex optimization problem (6.7), we run it several times and select the learners resulting from the highest log-likelihood L_C.

Algorithm 4 Expectation-Maximization algorithm for Mix-cFGLS utility learning of player i

function EM-MIX-CFGLS(X,Y,M,C_{limit})
  Initialization: Fit data with cFGLS learner using
  an appropriate Heteroskedasticity-noise structure G
  $\hat{G}_{EM} \leftarrow \hat{G}_{cFGLS}$ ▷ assignment of noise matrix
  $\theta_i^s \leftarrow \theta_{cFGLS}$ for $s = 1, \cdots, M$ ▷ θ_i^s initialization
  $\xi_i^s \leftarrow 0$ for $s = 1, \cdots, M$ ▷ ξ_i^s initialization
  $C \leftarrow C_{limit}$ ▷ convergence tolerance
  $k \leftarrow 1$ ▷ iteration number
  $M_{max} \leftarrow N$ ▷ upper iterations bound
  $L_I \leftarrow L_C$ ▷ Initial log-likelihood value using (6.10)
  Main Program:
while $k < M_{max}$ **do**
  Update latent variables τ_i using (6.11) ▷ *E–step*
  Update θ_i^{mix} solving (6.12) ▷ *M–step*
  Update ξ_i solving (6.13) ▷ *M–step*
  Update log-likelihood, L_N, using (6.10) **if** $L_N - L_I < C$ **then**
    break **else**
    $L_I \leftarrow L_N$
  $k \leftarrow k + 1$
  Outputs: θ_i^{mix}, ξ_i and L_N
end function

6.2 Forecasting via One-Level Hierarchical Mixture Model

Let's present the results of Mix-cFGLS utility learning applied to the data collected from the social game experiment 4.4. The forecasting using the proposed method is more accurate than ensemble utility methods such as bagging 5.4, and classical methods such as OLS. For the Mix-cFGLS we use two mixture components, one aggressive θ_A and one defensive θ_D, to represent each occupant. In particular, θ_A represents a player's profile that cares more about rewards, i.e. sacrifices comfort level for winning more points, while θ_D represents the opposite, i.e. covets comfort over winning.

Using occupant voting data we simulate the game defined by the learned Mix-cFGLS utility functions and show that the estimated model significantly reduces prediction error as compared to classical OLS and outperforms the bagged cFGLS forecast (see Table 6.1).

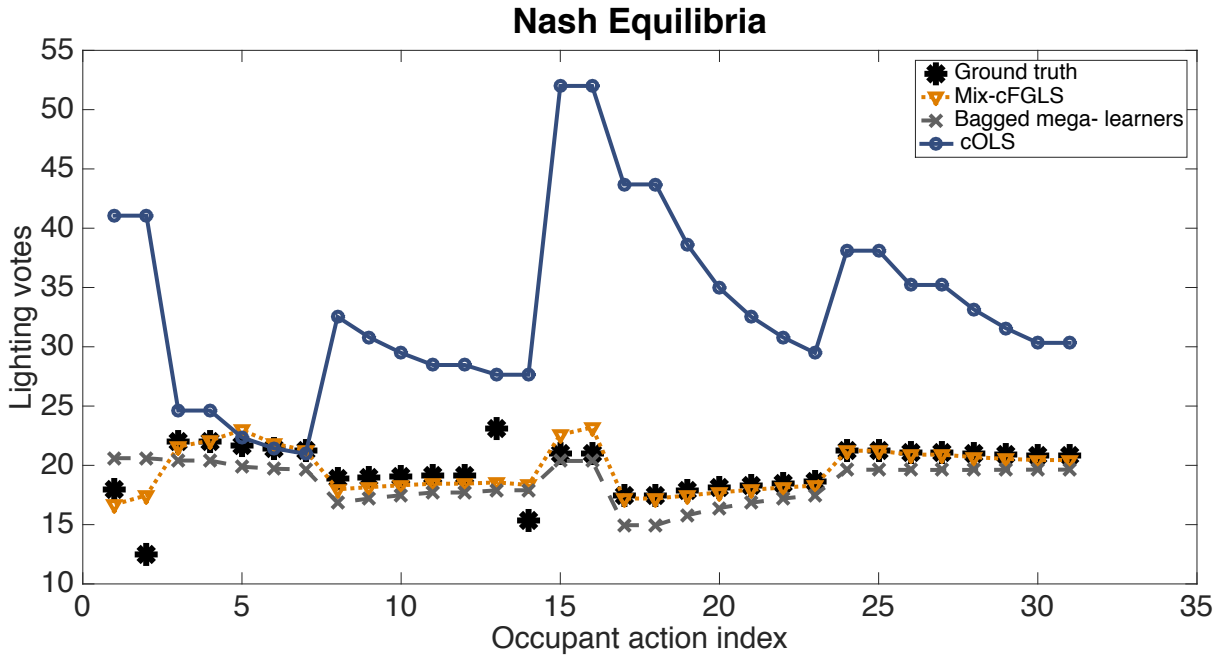


Figure 6.2: The ground truth mean of the observed lighting votes for default lighting setting of 20 is depicted by the black dots. The forecasting results via simulation of the occupant game using the cOLS, bagged mega-learners, and Mix-cFGLS learners are indicated in blue, grey, and orange respectively. On the x–axis we indicate the index of when a choice was made by one or more of the occupants (i.e. when the implemented lighting setting is changed); the time from one index to the next may be several minutes to hours depending on the activity level of the occupants. Notice that the mean of the Nash equilibria of the simulated game using the bagged mega-learners and Mix-cFGLS learners is approximately near the true mean where the cOLS learners produce Nash equilibria with a large error. The Mix-cFGLS learners have a nearly perfect forecast.

Our training data consisted of 80% of the users' actions in each default area. We test—i.e. compare the simulated forecast from our learned utility functions to the ground truth—on the remaining data for each default region. We employ cross-validation to prevent over-fitting.

In Figure 6.2, we see that the mixture of utilites model nearly approximates the ground truth and outperforms the other methods. In addition, using wild bootstrapping, we approximate the bias of the learners for the bagged cFGLS method. We remark that the Mix-cFGLS method aims to increase the bias in exchange for a reduction in variance. In Figure 6.3, we show the histogram of the cFGLS learners for player with user–id 2 obtained by replicates of data using wild bootstrapping. This particular occupant represents a player that prefers *comfort* to winning the majority of the time. Notice player's bagged cFGLS estimator is almost completely unbiased.

Table 6.1: Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Scaled Error (MASE) [65] of forecasting using Mix-cFGLS, bagged, and cOLS utility learners. Forecasting predicts occupants' behavior for default lighting settings of 20 and 10.

Default 20	**Mix-cFGLS**	Bagged cFGLS	cOLS
RMSE	**7.45**	8.31	22.53
MAE	**4.11**	5.20	18.35
MASE	**1.65**	2.08	7.34
Default 10	**Mix-cFGLS**	Bagged cFGLS	cOLS
RMSE	**7.75**	8.17	17.63
MAE	**6.42**	7.01	14.24
MASE	**4.42**	4.82	9.13

The Mix-cFGLS learner varies inside the grey region due to the fact that the softmax function gives a weighted sum of θ_A (red dotted vertical line) and θ_D (green dashed vertical line) learners (borders of the shaded region). This shows how the Mix-cFGLS introduces bias in the player's parameters estimates and, in exchange, we get more accurate forecasting of the users' actions. In machine learning, you can enhance forecasting accuracy by allowing for a small amount of bias if it results in a large reduction in variance. This is widely used in Ridge regression and in Lasso [46] in a form of a *prior* knowledge. In the Mix-cFGLS framework, we are able to explore the tradeoff between minimizing bias and variance. Indeed, having captured the noise structure using heteroskedasticity inference we are able to reduced estimation bias. However, since the learners in Mix-cFGLS framework are not static—they depend on historical data—we allow an amount of *bias* in order to gain a substantial decrease the *variance*.

6.3 Chapter Summary

A new hierarchical probabilistic utility learning framework is introduced. The new framework of parametric utility learning using a probabilistic interpretation for combining multiple utility functions via Mix-cFGLS uses a non-spherical noise model. The developed framework allows for the estimated parameters of the learned utility functions to depend on the actions of the players which, in turn, captures the fact that players' utility functions are not static and change based on environmental conditions. Moreover, Mix-cFGLS improves forecasting accuracy by allowing for a slight amount of bias in the utility learners in exchange for a reduction in variance. In particular, we show the theoretical Bias-Variance tradeoff in a player's unbiased profile where the utility learner based on our framework is able to reduce the overall forecasting error. Moreover, we applied the mixture of utilities method to learn the utility functions of participants in a social game 4.4 for inducing more efficient consumption of shared resources in smart buildings and showed that the forecasting error of the building

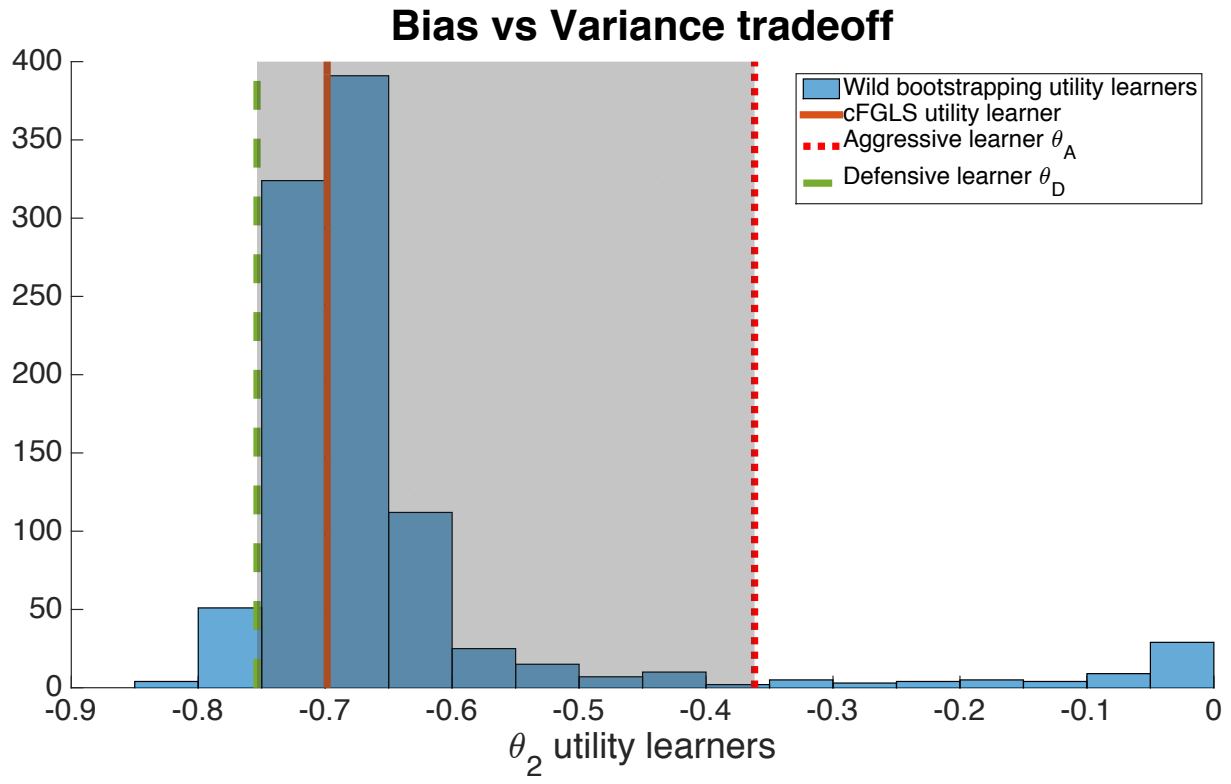


Figure 6.3: The histogram depicts the learners using the wild bootstrapping technique. We have approximately a Gaussian distribution around the initial cFGLS estimator which depicts an unbiased cFGLS utility learner. The vertical orange line represents the cFGLS estimator. The grey shaded region represents the area between the aggressive (red dotted vertical line) and defensive (green dashed vertical line) utility learners—i.e. θ_A and θ_D, respectively—estimated using Mix-cFGLS framework.

occupants' actions is significantly less than other methods such cOLS and bagged cFGLS. The mixture of utilities framework can be applied broadly to many different applications.

This framework enables us to close the loop around the building occupant and in effect, vary their behavior in order to meet, for instance, the requirements of a demand response program or to simply reduce over all consumption. Furthermore, such a framework is agnostic to the particular problem of energy efficiency in buildings and we believe it can be applied to any social game and could provide a useful tool in many experimental setups in smart city applications where learning decision-making behavior is crucial.

There are several directions for future research of the probabilistic representation of utility functions. One approach is by employing prior knowledge—e.g. survey results—into the estimation of players' utility functions. Incorporating survey data will result in a penalized Mix-cFGLS model and has the potential to lead to better forecasting. In addition, by exploring a graphical model representation for utility learning using Hidden Markov

Models (HMM) [18], more advanced Contextual Hidden Markov Model (CHMM) [34, 41, 164], or Hidden Conditional Random Fields (HCRF's) [139] with regression emissions we can efficiently model sequentially learning dependencies.

Chapter 7

From Correlations to Coalitions: Leveraging Network Effects in Utility Learning

In previous Chapters we have explored several utility learning frameworks—algorithms which result in extraordinary forecasting accuracy but in exchange of a significant run time complexity. In this Chapter, we propose a new method of data-driven modeling of human decision-making that reduces the complexity of existing methods while increasing the forecasting accuracy by accounting for possible correlations between players and form coalitions between agents. Building on existing game theoretic concepts such as *coalition games* [133], we are able to extend our existing robust utility learning framework 5.1 to an utility learning framework that has the potential to leverage interactions amongst players in order to reduce complexity and even operate online thereby allowing for adaptive incentive mechanisms to be implemented.

Specifically, we present two approaches for leveraging correlations in learning the utilities of non-cooperative agents' competing in a game: *correlation and coalition utility learning*. In both methods, we estimate the correlations between agents using constrained Feasible Generalized Least Squares with noise estimation 5.1. Hence, we use a small amount of training data to estimate the correlations between players and form coalitions between agents that are positively—negatively correlated. Using a small amount of data both methods run as constrained ordinary least squares (cOLS) amortize time. Amortized analysis [30] is used for analyzing *correlation and coalition utility learning* algorithms' complexity and the minimal usage of initial data helps derived algorithms to run with constrained ordinary least squares (cOLS) complexity.

In the *correlation utility learning*, we use the estimated correlations to generate a correlation utility function for each agent which is a weighted sum of its own estimated utility function and all the agents' estimated utilities that are highly correlated with them. We then optimize the weights to boost the performance of the estimators. In the *coalition utility learning*, we form coalitions between agents that are positively correlated. We then esti-

mate the parameters of the utility functions for each coalition where agents in a coalition jointly optimize their utilities. The correlation utility learning method outperforms existing schemes while the coalition utility learning method is simple enough to be adapted to an online framework after an initial training phase, yet it matches the performance of much more complex schemes. Most interestingly, both these correlation–based utility learning schemes are simple enough to be adapted to an online framework, yet they match the performance of much more complex estimation schemes.

By leveraging learned correlations between players we construct a correlated—coalition utility learning framework that matches the robust utility learning 5.1 errors while also being amenable to online implementation. The latter is important for integrating the proposed utility learning techniques with adaptive control or online incentive design. For example, it has been shown that static programs for encouraging energy efficiency are subject to the rebound effect in which participants often return to less efficient behavior after some time [107, 160]. By integrating our utility learning framework with incentive design, we will be able to create an adaptive model that learns how users' preferences change over time and thus, generate the appropriate incentives to ensure active participation.

Section 7.1 introduces the main idea—core framework of a correlation & coalition game under a Nash strategy. Building on top of this abstraction, in Section 7.2 we represent the main learning scheme while in Section 7.3 we demonstrate the new learning methods in smart-building social game data.

7.1 Decision-Making Model For Agents: Correlation & Coalition Games

Let's assume that agents' are utility maximizers and extend Nash non-cooperative game in *correlation & coalition game*. Then, we can extend base utility estimation framework 3.2 and adapt the new derived decision-making models.

7.1.1 Nash Play

A p-player game is described in terms of the strategy spaces and utility functions for each player. We denote by $\mathcal{I} = \{1, \dots, p\}$ the index set for players. Let $E_i^{m_i}$ denote the Euclidean strategy space of dimension m_i for player i and $x_i \in E_i^{m_i}$ denote its strategy vector. Define $m = \sum_i m_i$ and denote by $E^m = E_1^{m_1} \times \cdots \times E_p^{m_p}$ the joint strategy space and $x = (x_1, \dots, x_p)$ the joint strategy. Each player's strategy vector x_i is constrained to a convex set $S_i \subset E_i$. Let ℓ_i be the number of constraints on player i's problem and let $\ell = \sum_{i=1}^{p} \ell_i$. Denote by $S = S_1 \times \cdots \times S_p$ the constraint set which we can explicitly characterize in terms of mappings $h_i : E^{m_i} \to E^{\ell_i}$ where each component $h_{ij}(x)$, $j = 1, \dots, \ell_i$ is a concave function of x_i: $S_i = \{x_i | h_i(x_i) \geq 0\}$. It is assumed that S_i is non-empty and bounded.

Agent Optimization Problem. We model agents as *utility maximizers*—that is, the i–th player faces the optimization problem given by

$$\max_{x_i \in S_i} f_i(x_i, x_{-i}) \tag{7.1}$$

where $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p)$ is the marginal strategy vector for all players excluding player i.

Agent Interaction as a Game. In this framework, the agents are non-cooperative players in a continuous game with convex constraints. The outcome of their interaction is modeled using the Nash equilibrium concept:

Definition 4 (Nash Equilibrium)**.** *A point $x \in S$ is a* Nash equilibrium *for the coalition game $(f_1, \ldots, f_n)$ on S if for each $i \in \{1, \ldots, p\}$ $f_i(x_i, x_{-i}) \geq f_i(x_i', x_i)$, $\forall\ x_i' \in S_i$.*

It is well known that Nash equilibria exist for concave games [152]. The notion of a Nash equilibrium can be relaxed in the following way:

Definition 5 (ε–Approximate Nash Equilibrium)**.** *A point $x \in S$ is a* Nash equilibrium *for the coalition game $(f_1, \ldots, f_n)$ on S if for each $i \in \{1, \ldots, p\}$ $f_i(x_i, x_{-i}) \geq f_i(x_i', x_i) - \varepsilon$, $\forall\ x_i' \in S_i$.*

7.1.2 Nash Play: Coalition Games

When incentivized to do so—e.g., because they stand to increase their payoff—players form coalitions in which members of the coalition jointly optimize their utilities. That is, the set of players $\mathcal{I}$ is partitioned into subsets such that players in each subset collude.

Suppose the set of players $\mathcal{I}$ is partitioned into p_c coalitions. We will use the notation $\mathcal{C}_i$ as the index set for coalition i for each $i \in \{1, \ldots, p_c\}$. Then, players in coalition $\mathcal{C}_i$ seek to solve the optimization problem given by

$$\max_{x_{\mathcal{C}_i} \in S_{\mathcal{C}_i}} f_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) \tag{7.2}$$

where $f_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) = \sum_{j \in \mathcal{C}_i} f_j(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i})$, $S_{\mathcal{C}_i} = \times_{j \in \mathcal{C}_i} S_j$, $x_{\mathcal{C}_i} = (x_j)_{j \in \mathcal{C}_i}$ and $x_{-\mathcal{C}_i} = (x_j)_{j \in \mathcal{I}/\mathcal{C}_i}$.

In our framework, we will assume that utilities are transferrable—that is players can losslessly transfer part of its utility to another player. As an example, players in a coalition may agree to divide the payoff equally or may agree to some alternative distribution of the payoff in a side contract which we leave unmodeled. In general, players are incentivized to participate in a coalition if the utility of participating is greater than if they played the game as a selfish individual.

Again, their interaction is modeled using the Nash equilibrium concept:

Definition 6 (Nash Equilibrium)**.** *A point $x \in S$ is a* Nash equilibrium *for the coalition game $(f_1, \ldots, f_n)$ on S if for each $i \in \{1, \ldots, p_c\}$ $f_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) \geq f_{\mathcal{C}_i}(x_{\mathcal{C}_i}', x_{-\mathcal{C}_i})$, $\forall\ x_{\mathcal{C}_i}' \in S_{\mathcal{C}_i}$.*

If $p_c = p$ and each player is in its own coalition by itself, then the above definition reduces to the definition of a Nash equilibrium for the p–player non-cooperative game $(f_1, \ldots, f_n)$ as defined in Definition 4.

The definition can be similarly relaxed as follows:

Definition 7 (ε–Approximate Nash Equilibrium)**.** *Given $\varepsilon > 0$, a point $x \in S$ is a ε–approximate* Nash equilibrium *for the coalition game $(f_1, \ldots, f_n)$ on S if for each $i \in \{1, \ldots, p_c\}$ $f_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) \geq f_{\mathcal{C}_i}(x'_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) - \varepsilon$, $\forall\ x'_{\mathcal{C}_i} \in S_{\mathcal{C}_i}$.*

7.2 Utility Learning Under Correlation & Coalition Games

In this section, we develop the main learning frameworks given an abstraction of correlation or coalition game.

7.2.1 Wild Bootstrapping: Asymptotic Approximation of Network Effects

When the number of unknown parameters is large compared to the size of the observation set—which is the case for the social game application we present in Section 4.4 as well as many other applications where human decision-makers are involved—*wild bootstrapping* [44, 46] can be used to generate a *psuedo-data set*. Wild bootstrapping is a technique of parametric bootstrapping that is consistent with heteroskedastic inference and a cFGLS data generation process. In general, bootstrapping is a technique for asympotic approximation of the bias and standard error of an estimator in a complex and noisy statistical model.

The bootstrapping process can be described in two steps:

First, we fit our cFGLS model which gives us $\hat{\beta}_{\text{cFGLS}}$. For cFGLS regression step, we use Freedman noise structures as is described in Section 5.1. More specifically, let $\hat{\beta}^{\text{cOLS}}$ be the cOLS estimate of β with residual vector $e = Y - X\hat{\beta}^{\text{cOLS}} \in \mathbb{R}^{n_o}$. The residual vector e can be decomposed into residuals for each player by writing $e = [e_1^\top \ \cdots \ e_p^\top]^\top$. We use e_i to compute an estimate $\hat{K}_i$ of K_i which is, in turn, used to compute $\hat{G}$. The residuals come in ℓ_i pairs since at each observation k, $Y_i^{(k)} \in \mathbb{R}^{\ell_i+1}$. For ease of presentation and comprehension, we will use a paired index for the residuals instead of a single index. There are n_i instances at which we have $\ell_i + 1$ observations. Let $(e_i)_{k,j} = (e_i)_{(\ell_i+1)(k-1)+j}$ where $k \in \{1, \ldots, n_i\}$ and $j \in \{1, \ldots, \ell_i + 1\}$. Then, with the residuals, we form estimates $\hat{B}_{i,k} \in \mathbb{R}^{(\ell_i+1)\times(\ell_i+1)}$ of $B_{i,k}$ using $(\hat{B}_{i,k})_{jj} = n_i^{-1} \sum_{t=1}^{n_i} e_{t,j}^2$ and $(\hat{B}_{i,k})_{lj} = n_i^{-1} \sum_{t=1}^{n_i} e_{t,j} e_{t,\ell}$ for $j \neq \ell$ in

$$\hat{B}_{i,k} = \begin{bmatrix} (\hat{B}_{i,k})_{11} & \cdots & (\hat{B}_{i,k})_{1(\ell_i+1)} \\ \vdots & \ddots & \vdots \\ (\hat{B}_{i,k})_{(\ell_i+1)1} & \cdots & (\hat{B}_{i,k})_{(\ell_i+1)(\ell_i+1)} \end{bmatrix} \tag{7.3}$$

Then, in the second step we generate N replicates of *pseudo–data* using the data generation process

$$\tilde{Y} = X\hat{\beta}_{cFGLS} + \Phi(e)\varepsilon, \tag{7.4}$$

where $\tilde{Y} \in \mathbb{R}^{n_d \times 1}$ is the new observation vector (pseudo-observations), $\hat{\beta}_{cFGLS} \in \mathbb{R}^{n_d \times 1}$ is the cFGLS estimator, $\varepsilon \sim N(0, I^{n_d \times n_d})$, $e \in \mathbb{R}^{n_d \times 1}$ is the residual vector given by $e = \tilde{Y} - X\hat{\beta}_{cFGLS}$ and $\Phi : \mathbb{R}^{n_d \times 1} \to \mathbb{R}^{n_d \times 1}$ is a nonlinear transformation such that $\Phi(e) = \hat{G}^{\frac{1}{2}} \in \mathbb{R}^{n_d \times n_d}$. Since $E(\Phi(e)\varepsilon|X) = \Phi(e)E(\varepsilon|X) = \Phi(e)E(\varepsilon) = 0_{n_d \times n_d}$, using the data generation process in (7.4), we re-sample from i.i.d variables.

Using Wild bootstrapping, the empirical covariance matrix of β^*_{est} and the average of $\beta^*_{est} - \beta_{est}$ are asymptotic approximations of the covariance matrix and bias, respectively. Asymptotic estimation of the empirical covariance matrix reveals hidden structures between agents and is what we leverage both in the correlation and coalition utility learning procedures.

7.2.2 Correlated Utility Learning

In this section, we describe how we leverage learned approximated correlations between agents to *boost* our estimators. In particular, the empirically learned correlations are used to reduce the forecasting error by crafting a new correlated game in which we construct a *correlation utility* for each player by composing a weighted sum of the player's estimated utility and the estimated utilities of all the players that are highly correlated with it. We then optimize over the weights in order to further reduce the forecasting error.

When the correlations between players are positive, we create what we refer to as *psuedo-coalitions* since players are not *explicitly* agreeing to collude in the game but rather are doing so *implicitly*. The degree of *psuedo-coalition* is discovered by the robust utility learning process through estimating the empirical covariance of $\hat{\beta}^{\text{cFGLS}}$.

On the other hand, when the correlations between players are negative, we find that these negative correlations can be used to take advantage of active players' richer data sets in predicting the behavior of players that less active or ones that have little variation in their data.=

We refer to the learned utility—$\hat{f}_i$ for player i—from the robust utility learning framework as the *nominal utility* whose estimate is given by

$$\hat{f}_i(x_i, x_{-i}) = \varphi_{i,0}(x_i, x_{-i}) + \textstyle\sum_{j=1}^{N_i} \hat{\theta}^{\text{cOLS}}_{i,j} \varphi_{i,j}(x_i, x_{-i}) \tag{7.5}$$

where $\hat{\theta}^{\text{cOLS}}_i$ is extracted from the cOLS estimated $\hat{\beta}^{\text{cOLS}}_i = (\hat{\mu}^{\text{cOLS}}_i, \hat{\theta}^{\text{cOLS}}_i)$ which represents our estimate using cOLS.

Using the correlations we learn when we estimate $\hat{\beta}^{\text{cFGLS}}$ as described above, we construct a new utility $\hat{g}_i$ by combining scaled versions of a subset (potentially all) of the other players' utilities that are correlated with player i. Let $\mathcal{Q}_i \subset \mathcal{I}$ denote the subset of players correlated with player i and let $\mathcal{K}_i \subset \mathcal{Q}_i$ be the set of players used in constructing $\hat{g}_i$. The correlated

utility $\hat{g}_i$ for player i is given by

$$\hat{g}_i(x) = \sum_{l \in \mathcal{Q}_i} \left(\alpha_{i,l} c_{i,l} \varphi_{i,0}(x) + \sum_{j=1}^{N_j} \alpha_{i,l} \hat{\theta}_{i,j}^{\text{cOLS}} \varphi_{i,j}(x) \right) \tag{7.6}$$

where as usual $x = (x_i, x_{-i})$, $\alpha_{i,i}$ is the estimated variance of player i determined by the empirical covariance matrix, $\alpha_{i,l}$ is the covariance between the parameter estimates for player i and l also determined by the empirical covariance matrix, and c_{il} are scaling constants over which we optimize.

We refer to the resulting game as an *approximated correlation game*[1].

Given the form of $\hat{g}_i$, our goal is to optimize the scaling constants c_{il} in order to reduce the forecasting error. We formulate a convex optimization problem using the first– and second–order conditions on each player's individual optimization problem where we assume that player i is now solving the problem given by

$$\max_{x_i \in S_i} \hat{g}_i(x_i, x_{-i}). \tag{7.7}$$

The convex optimization problem we solve is formulated in a similar fashion to the base utility learning problem of Section 3.2.

Let $c_i \in \mathbb{R}^{|\mathcal{K}_i|}$ be defined as $c_i = (c_{i,j})_{j \in \mathcal{K}_i}$ and let $c = (c_i)_{i \in \mathcal{I}}$.

Let the residual of the stationarity condition of (7.7) be given by

$$r_{\text{s},i}^{(k)}(z_i, \mu_i; \hat{\theta}_i^{\text{cOLS}}) = D_i \hat{g}_i(x^{(k)}) + \mu_i^\top D_i h_i(x_i^{(k)}) \tag{7.8}$$

and the residual of the complementary slackness conditions be given by

$$r_{\text{c},i}^{j,(k)}(\mu_i) = \mu_{i,j} h_{i,j}(x_i^{(k)}), \ j \in \{1, \ldots, \ell_i\}. \tag{7.9}$$

As before, let $r_{\text{c},i}^{(k)}(\mu_i) = [r_{\text{c},i}^{1,(k)}(\mu_i) \ \cdots \ r_{\text{c},i}^{\ell_i,(k)}(\mu_i)]$.

Define $Q_i \in \mathbb{R}^{n_i \times |\mathcal{K}_i|}$ by

$$Q_i = \left[\alpha_{i,j} D_{i,i}^2 \varphi_{i,0}(x^{(k)}) \right]_{k=1, j \in \mathcal{K}_i}^{n_i}. \tag{7.10}$$

and $q_i \in \mathbb{R}^{n_i}$ by

$$q_i = \left[\sum_{j \in \mathcal{K}_j} \alpha_{i,j} \left(\sum_{l=1}^{N_i} \hat{\theta}_{il}^{\text{cOLS}} D_{i,i}^2 \varphi_{i,l}(x^{(k)}) \right) \right]_{k=1}^{n_i}. \tag{7.11}$$

Then, we have the following convex optimization problem:

$$\begin{aligned} &\min_{c,\mu} \sum_{i=1}^{p} \sum_{k=1}^{n_i} \chi(r_{\text{s},i}^{(k)}(z_i, \mu_i; \hat{\theta}^{\text{cOLS}}), r_{\text{c},i}^{(k)}(\mu_i)) \\ &\text{s.t.} \ \ Q_i z_i + q_i \leq 0, \ \mu_i \geq 0 \ \ \forall \ i \in \mathcal{I} \end{aligned} \tag{P–2}$$

Solving (P–2) gives us estimated correlated utilities $\hat{g}_i$ for each $i \in \mathcal{I}$ that we then use to forecast the players' decisions.

[1] We remark that there exists an equilibrium concept called *correlated equilibrium* [7] which generalizes a Nash equilibrium by characterizing correlations between randomized strategies; we mention this only to alleviate any potential confusion. The equilibrium concept we utilize for the approximated correlation game is still a pure Nash equilibrium.

7.2.3 Coalition Utility Learning

As an alternative approach, when players are highly correlated we can re-estimate their parameters by returning to the utility learning procedure but now with players who are highly correlated treated as if they are participating in a coalition. Using the empirically learned correlations, we partition the set of players $\mathcal{I}$ into p_c coalitions. Analogous to the correlation utility learning method, our aim is to define a *coalition utility* $\tilde{g}_{\mathcal{C}_i}$ and estimate the parameters of the coalition utilities assuming coalitions play a game against each other, where those in a coalition jointly optimize their utilities.

Let $-\mathcal{C}_i = \mathcal{I}/\mathcal{C}_i$ be the set of players not in coalition $\mathcal{C}_i$. The coalition utility $\tilde{g}_{\mathcal{C}_i}$ for player i is given by

$$\tilde{g}_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) = \textstyle\sum_{j\in\mathcal{C}_i} f_j(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) \tag{7.12}$$

where the nominal utility function used for the coalition game for player i is given by

$$f_i(x) = \textstyle\sum_{j\in\mathcal{C}_i} \varphi_{j,0}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}) + \sum_{l=1}^{N_j} \varphi_{j,l}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i})\theta_{jl}$$

with $x = (x_{\mathcal{C}_i}, x_{-\mathcal{C}_i})$. Then players in $\mathcal{C}_i$ are jointly solving

$$\max_{x_{\mathcal{C}_i}\in S_{\mathcal{C}_i}} \tilde{g}_{\mathcal{C}_i}(x_{\mathcal{C}_i}, x_{-\mathcal{C}_i}). \tag{7.13}$$

Given $\tilde{g}_{\mathcal{C}_i}$, we develop a convex optimization problem to estimate parameters θ_i in order to reduce the forecasting error. Again, the problem is formulated in a similar fashion to the base utility learning problem of Section 3.2.

Define the vector $\theta_{\mathcal{C}_i} \in \mathbb{R}^{M_i}$ where $M_i = \sum_{j\in\mathcal{C}_i} N_j$ by $\theta_{\mathcal{C}_i} = \left(\theta_1, \theta_2, \ldots, \theta_{|\mathcal{C}_i|}\right)$. For optimization problem (7.13), let the residual of the stationarity condition be given by

$$r^{(k)}_{\mathrm{s},\mathcal{C}_i}(\theta_{\mathcal{C}_i}, \mu_i) = D_i\tilde{g}_i(x^{(k)}_{\mathcal{C}_i}, x^{(k)}_{-\mathcal{C}_i}) + \textstyle\sum_{j=1}^{|\mathcal{C}_i|} \mu_j^\top D_j h_j(x_j^{(k)}) \tag{7.14}$$

and the residual of the complementary slackness conditions be given by

$$r^{j,(k)}_{\mathrm{c},l}(\mu_l) = \mu_{l,j}h_{l,j}(x_l^{(k)}),\ j \in \{1, \cdots, \ell_l\},\ l \in \mathcal{C}_i. \tag{7.15}$$

with $r^{(k)}_{\mathrm{c},l}(\mu_l) = [r^{1,(k)}_{\mathrm{c},l}(\mu_l)\ \cdots\ r^{\ell_l,(k)}_{\mathrm{c},l}(\mu_l)]^\top$, $r^{(k)}_{\mathrm{c},\mathcal{C}_i}(\mu_{\mathcal{C}_i}) = [r^{(k)}_{\mathrm{c},l}(\mu_l)]^\top_{l\in\mathcal{C}_i}$ and $\mu_{\mathcal{C}_i} = (\mu_l)_{l\in\mathcal{C}_i}$. Let $n_{\mathcal{C}_i} = \sum_{j\in\mathcal{C}_i} n_j$.

To estimate the coalition utilities, we solve

$$\begin{aligned} &\min_{\theta,\mu} \textstyle\sum_{i=1}^{N_c} \sum_{k=1}^{n_{\mathcal{C}_i}} \chi(r^{(k)}_{\mathrm{s},\mathcal{C}_i}(\theta_{\mathcal{C}_i}, \mu_{\mathcal{C}_i}), r^{(k)}_{\mathrm{c},\mathcal{C}_i}(\mu_{\mathcal{C}_i})) \\ &\text{s.t.}\ \ \theta_j \in \Theta_j,\ \mu_j \geq 0\ \ \forall\ j \in \mathcal{I} \end{aligned} \tag{P–3}$$

Solving (P–3) using the ℓ_2–norm for the convex penalty function χ gives us an ordinary least squares framework for estimating coalition utilities $\tilde{g}_{\mathcal{C}_i}$ for each $i \in \{1, \ldots, p_c\}$.

7.3 Forecasting via Correlation & Coalition Utility Learning

We now present the results of the proposed framework for leveraging correlations between player actions. We apply the correlation boosted utility learning methods—correlation and coalition utility learning—to data collected from the social game experiment described in Section 4.4. The social game data set is composed of lighting votes participants made throughout the duration of the experiment. The time from one vote to the next may be several minutes to hours depending on the activity level of the participants.

For both methods, we select combinations of players in support of improving the estimators' performance by utilizing information learned from players with the most variation in their votes in order to improve the estimates of players who consistently vote the same value or have a limited participation record. In this way, we *boost* the performance of our utility learning scheme by transferring information providing by the voting record of the more active players to other players.

For the correlation utility learning method, we apply a 10–fold cross validation [46] procedure with an 80%–20% training/testing data split in order to limit over-fitting. We use a small subset of the training data (approximately 3–5% of the data which is roughly 2 days of the experiment) to approximate the correlations between players. For the estimation of the correlations between players we use the robust utility learning method described in Section 5.1 and specifically leveraging wild parametric bootstrapping 7.2.1, which gives us $\hat{C}_\beta$. In Table 7.1, we show a subset of the estimated covariance matrix $\hat{C}_\beta$. Using these values, we construct a correlated game as described in Section 7.2.2 for which we estimate the parameters using the correlated utility learning method.

We construct a correlation game with the following *pseudo-coalitions*:

1. $\mathcal{K}_2 = \{2, 6, 20\}$: player 2's utility function is modified by player 6's and player 20's where each of these players are what we call *passive players* (i.e. their votes tend to be strongly related to their satisfaction as opposed to increasing their chances of winning—see the red cells in Table 7.1);

2. $\mathcal{K}_8 = \{8, 14\}$: player 8's utility function is modified by player 14's where player 8 and 14 are what we call *aggressive players* (i.e. their votes tend to be much lower indicating a greater desire to win points—see the green cells in Table 7.1);

3. $\mathcal{K}_{14} = \{2, 8, 14\}$.: player 14's utility function is modified by player 8's and player 2's where player 14 is positively correlated with player 8 and negatively correlated with player 2—see the blue cells in Table 7.1.

All other players' utilities in the correlated game remain unchanged; that is, they are taken to be $\hat{g}_i \equiv \hat{f}_i$, $i \in \mathcal{I}/\{2, 8, 14\}$. We use the cOLS estimated parameters $\hat{\theta}_i^{\text{cOLS}}$ to create the correlation game. Then we apply the correlation utility learning method to optimize the $c_{i,j}$'s.

Table 7.1: Estimated covariance matrix for the most active players. The colored column-row pairs indicate the agents used to create the correlation game—i.e. the column indicates the agent whose estimated parameter is used to modify the row agent's utility.

Id	**2**	**6**	**8**	**14**	**20**
2	0.04	**0.06**	-2.80	-5.19	**0.03**
6	0.06	7.84	-16.8	0.84	-0.02
8	-2.80	-16.8	6.4×10^4	$\mathbf{4.28\times10^4}$	-7.60
14	**-5.19**	0.84	$\mathbf{4.28\times10^4}$	8.84×10^4	-12.6
20	0.03	-0.02	-7.60	-12.6	0.07

For the coalition utility learning method, we again divide the data into training and testing data sets with an 80%–20% split (and use cross validation). As in correlation utility learning we use a small subset of the training data (approximately 3–5% of the data which is roughly 2 days of the experiment) to approximate the correlations between players. With these correlations we select coalitions. Then we use cOLS to estimate the parameters of the utilities for the coalition game.

The reason we use only a small subset of the data is that the cFGLS and noise estimation scheme described in Section 5.1 is computationally expensive, especially with a larger bootstrapped data set. Our ultimate goal is to have a utility estimation method with low forecasting error that is simple enough to be converted to an online estimation scheme so that it can be integrated into an adaptive incentive design algorithm [147]. By employing cOLS in the estimation step, we are able to partially meet this goal.

Using the social game data, we select the most correlated players which happen to be players 8 and 14—the estimated correlation between these players is several orders of magnitude greater than the correlation between any of the other players. Hence, we create the coalition game $\mathcal{G}_{\text{coal}} = \{\tilde{g}_{\mathcal{C}_1}, \tilde{g}_{\mathcal{C}_2}\}$ where $\mathcal{C}_1 = \{8, 14\}$ and $\mathcal{C}_i = \{i\}$ for $i \in \mathcal{I}/\mathcal{C}_1$. As we noted, players 8 and 14 are both aggressive players. However, player 8 has little variation in its voting record—the voting record contains mostly zero votes. One interpretation is that the correlation method is in some sense serving to reduce dimensionality since players are clustered together and *shared information* between them is leveraged to improve the forecast performance.

In Table 7.2, we present the root mean square error (RMSE), mean average error (MAE), and mean absolute scaled error (MASE) for the cOLS and cFGLS estimators and the estimated correlated utilities $\{\hat{g}_i(\cdot; \{\hat{\theta}_j^{\text{cOLS}}\}_{j\in\mathcal{K}_i})\}_{i\in\mathcal{I}}$ and coalition utilities $\{\tilde{g}_{\mathcal{C}_i}(\cdot; \{\tilde{\theta}_j^{\text{coal}}\}_{j\in\mathcal{C}_i})\}_{i\in\mathcal{I}}$ (where the $\tilde{\theta}_j^{\text{coal}}$'s are re-estimated using cOLS as in (P–3)). In the lower plot of Figure 7.1, we show the forecast produced by the cOLS, cFGLS, correlated, and coalition utility learning methods. We see that the correlated and coalition estimation schemes reduce the estimation error when comparing to cOLS. Their performance is on par with cFGLS and the correlated estimation scheme even outperforms cFGLS.

In general, cOLS performs poorly when players are treated as selfish individuals (see the

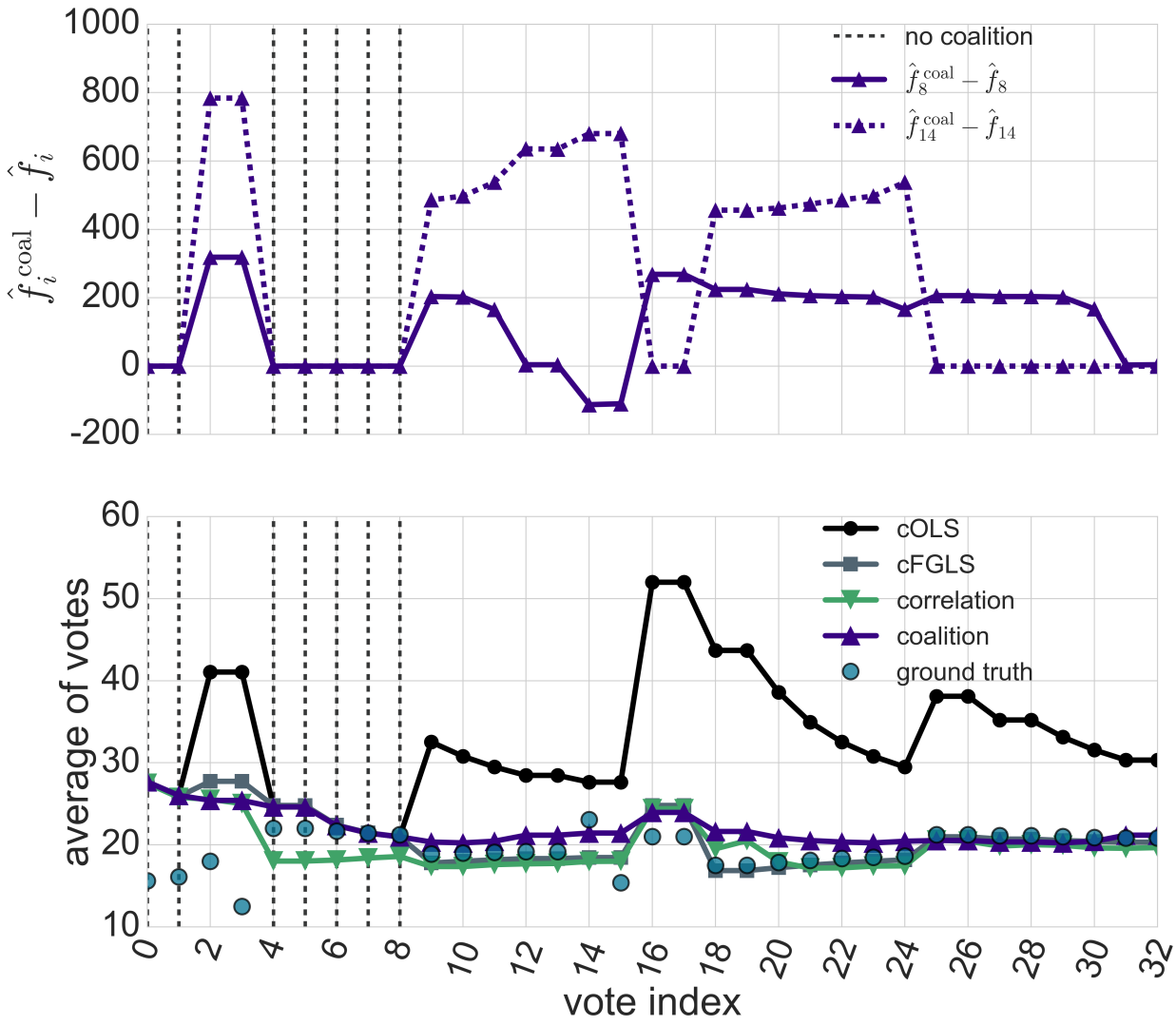


Figure 7.1: Forecasting results for default lighting setting 20 (lower plot) and happiness metric comparing estimated utilities using the coalition $\hat{f}_i^{\text{coal}}$ and cOLS $\hat{f}_i$ utility learning methods (upper). The x–axis values indicate the index of when a choice was made by one or more of the occupants (i.e. when the implemented lighting setting is changed); the time from one index to the next may be several minutes to hours depending on the activity of the participants. The **dark gray** dashed lines indicate when no coalition was used in the coalition estimate (instead all players played selfishly—this occurs when player's 8 and 14 are not both present in the office and thus, cannot collude).

lower plot in Figure 7.1 and Table 7.2)—this is in part due to the size and lack of variation of the votes. Yet, using the correlation estimation scheme the cOLS estimators performance is improved by optimizing the weights of the correlation utilities. The coalition estimation

Table 7.2: Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Scaled Error (MASE) for the forecast using the cOLS, cFGLS, and correlation, and coalition utility learning methods in the default lighting setting 20.

Error	cOLS	cFGLS	correlated	coalition
RMSE	22.53	11.36	11.3	12.79
MAE	18.35	6.81	6.49	7.45
MASE	7.34	2.72	2.63	3.02

scheme also reduces the estimation as compared to cOLS. This is, again, due to the fact that the method allows for *information sharing* since data from one player is used in estimating the utility of another player via learned correlations.

Moreover, in the upper plot of Figure 7.1, we show the difference between the coalition estimated utility and the cOLS estimated utility for players 8 and 14. In particular, using each observed approximate Nash $x^{(k)}$, we compute the cOLS estimated utility value

$$\hat{f}_i(x^{(k)}; \hat{\theta}_i^{\text{cOLS}}) \tag{7.16}$$

and the coalition estimated utility value which we take to be an equitable distribution of the payoff amongst the players in a coalition, i.e.

$$\hat{f}_i^{\text{coal}}(x^{(k)}; \hat{\theta}_i^{\text{coal}}) = \tfrac{1}{|\mathcal{C}_1|}\tilde{g}_{\mathcal{C}_1}(x^{(k)}; \hat{\theta}_{\mathcal{C}_1}^{\text{coal}}) \tag{7.17}$$

where $\hat{\theta}_{\mathcal{C}_1}^{\text{coal}} = (\hat{\theta}_i^{\text{coal}})_{i\in\mathcal{C}_1}$ and each $\hat{\theta}_i^{\text{coal}}$ are the cOLS estimated parameters for the coalition game. We then compute the *happiness metric* which we define to be the difference between these two estimated utilities:

$$\hat{f}_i^{\text{coal}}(x^{(k)}; \hat{\theta}_i^{\text{coal}}) - \hat{f}_i(x^{(k)}; \hat{\theta}_i^{\text{cOLS}}). \tag{7.18}$$

We remark that it is difficult to estimate the true structure of coalition side payments/utility transfers were they actually taking place. What is interesting, however, is that not only do we see improved estimator performance but also the happiness metric—which assumes a uniform distribution of wealth amongst coalitions—indicates that in fact the players have greater utility when treated as colluding. Specifically, looking at Figure 7.1, when there is no coalition (dashed grey lines), the utilities are equal which should be the case; however, when we use a coalition for players 8 and 14, they are *better off* under the coalition utility with the exception of player 8 on two voting instances (votes indexed by 14 and 15). The fact that the players are generally happier under the coalition estimated utility than the cOLS estimated utility may indicate that there is some (explicit or implicit) collusion happening in practice.

7.4 Chapter Summary

We developed two novel utility learning schemes that leverage estimated correlations between players in order to boost the performance of the estimated utilities in forecasting player decisions. Both methods outperform existing techniques based on classical estimation methods. Moreover, the coalition utility learning method is significantly less computationally intensive than cFGLS, which is introduced in Section 5.1. After an initial batch training phase to compute correlations, it is amenable to online implementation and thus, has the potential to be integrated into an online algorithm for utility learning and incentive design.

We remark that the incentive mechanisms will ultimately modify players' utilities and thus, whether or not they are incentivized to collude, a central planner (such as a building manager or, more broadly, a service provider). This exposes an interesting avenue for future research in investigating the persistence of equilibria or coalitions after introduction of incentives.

Chapter 8

Utility Learning Under Discrete Choice Games: A Deep Learning Approach

The implementation of smart building technology in the form of smart infrastructure applications has great potential to improve sustainability and energy efficiency by leveraging humans-in-the-loop strategy. Adoption of human-centric building services and amenities also leads to improvements in the operational efficiency of cyber-physical systems that are used to control building energy usage. However, human preference in regard to living conditions is usually unknown and heterogeneous in its manifestation as control inputs to a building. Furthermore, the occupants of a building typically lack the independent motivation necessary to contribute to and play a key role in the control of smart building infrastructure. Moreover, true human actions and their integration with sensing/actuation platforms remains unknown to the decision maker tasked with improving operational efficiency.

In Chapter 3 a base utility learning framework is introduced while in Chapters 4, 5, 6, and 7 advanced learning techniques are developed for non-cooperative continuous games assuming the humans play according to a Nash strategy. However, in the current Chapter we want to answer the following questions:

How can we enable and motivate humans in a gamification setting in which they don't immediate compete with others for a shared source? How can we develop a gaming framework in which players' co-optimize their own utility functions across a set of discrete & mutually exclusive actions? Then, what is the affect of possible large data-sets of players' actions and (or) features?

By modeling user interaction as a *sequential discrete game* between non-cooperative players, we introduce a gamification approach for supporting user engagement and integration in a human-centric cyber-physical system. We propose the design and implementation of a large-scale network game with the goal of improving the energy efficiency of a building

through the utilization of cutting-edge Internet of Things (IoT) sensors and cyber-physical systems sensing/actuation platforms. By observing human decision makers and their decision strategies in their operation of building systems, we can apply inverse learning techniques in order to estimate their utility functions. Game theoretic analysis often relies on the assumption that the utility function of each agent is known a priori; however, this assumption usually does not hold in many real-world applications.

Our framework is centered around learning agent preferences over room resources such as lighting and A/C as well as external parameters like weather conditions, high-level grid control, and provided incentives. Specifically, we model decision-making agents as sequential utility maximizers. Agents are strategic entities that make decisions based on their own preferences without consideration of others. The game-theoretic framework both allows for qualitative insights to be made about the outcome of aforementioned selfish behavior—more so than a simple prescriptive model—and, more importantly, can be leveraged in designing mechanisms for incentivizing agents.

A benchmark utility learning framework is proposed that employs robust estimations for classical discrete choice models provided with high dimensional imbalanced data. To improve forecasting performance, we extend the benchmark utility learning scheme by leveraging Deep Learning end-to-end training with Deep bi-directional Recurrent Neural Networks. We apply the proposed methods to high dimensional data from a social game experiment designed to encourage energy efficient behavior among smart building occupants in Nanyang Technological University (NTU) residential housing. Using occupant-retrieved actions for resources such as lighting and A/C, we simulate the game defined by the estimated utility functions to demonstrate the performance of the proposed methods on ground truth data. For demonstrations of our infrastructure and for downloading de-identified high dimensional data sets, please visit our web site [1].

More concretely, we model decision-making agents as *utility maximizers* and, using machine learning techniques, we derive a benchmark & Deep Learning scheme to infer their utility functions. At the core of our approach is the decision to model the occupants as non-cooperative agents who play according to a sequential discrete choice game. Discrete choice models have been used extensively to examine modes of transportation [172], choice of airport [10], demand for organic foods [55], robbery patterns [11], and even school social interactions [165]. Under this assumption of non-cooperation, we were able to use a randomized utility framework and propose novel utility estimation algorithms for the occupants' utility functions. Most importantly, estimating agent utility functions via our method results in predictive models that provide excellent forecasting of occupant actions—usage. The game-theoretic framework both allows for qualitative insights to be made about the outcome of such selfish behavior—more so than a simple prescriptive model—and, more importantly, can be leveraged in designing mechanisms for incentivizing agents.

Moreover, we provide a demo web portal for demonstrating our infrastructure and for

[1] *smartNTU* demo web-portal: https://smartntu.eecs.berkeley.edu

downloading de-identified high dimensional data sets[2]. High-dimensional data sets can serve either as a benchmark for discrete choice model learning schemes or as a great source for analyzing occupants' usage in residential buildings. Towards this scope, we use conventional deep variational auto-encoders [94] or recurrent network based adaptation of variational auto-encoders [57] as an approach to create a nonlinear manifold (encoder) that can be used as a generative model. Variational auto-encoders can fit large high dimensional data sets (like our social game application) and train a deep model to generate data like the original data set. In a sense, generative models automate the natural features of a data set and then provide a scalable way to reproduce known data. This capability can be employed either in the utility learning framework for boosting estimation or as a general way to create simulations mimicking occupant behavior—preferences.

A series of experimental trials were conducted to generate real-world data, which was then used as the main source of data for our approach. This differentiates our work from a large portion of other works in the same field that use simulations in lieu of experimental methods. In many cases, participants exhibit a tendency to revert to previously inefficient behavior after the conclusion of a program. Our approach combats this effect by implementing incentive design that can adapt to the behavior and preferences of occupants progressively, which ensures that participants are continuously engaged. From a managerial perspective, the goal is to minimize energy consumption while maximizing occupant comfort. With this social game framework, the manager is capable of considering the individual preferences of the occupants within the scope of the building's energy consumption. The advent of this social game system could potentially offer an unprecedented amount of control for managers without sacrificing occupant comfort and independence.

The broader purpose of this Chapter is to introduce a general framework that leverages game-theoretic concepts to learn models of players' decision-making in residential buildings provided with our implementation of a novel energy social game. In Section 8.1 we introduce the gamification approach to energy conservation at Nanyang Technological University. The core gaming abstraction is introduced in Section 8.2. Then, we present a variety of benchmark utility learning models and a novel pipeline for efficient training in Section 8.3. Novel sequential decision-making models using end-to-end Deep Learning models focused on the utilization of deep recurrent neural networks for capturing gaming data are introduced in Section 8.4 while results of the proposed methods are highlighted at Section 8.5.

8.1 A Gamification Approach to Energy Conservation at Nanyang Technological University: A Smart Building Social Game

In this section, we introduce the concept of our social game implemented at Nanyang Technological University (NTU) residential housing apartments along with the software architecture

[2] *smartNTU* demo web-portal: https://smartntu.eecs.berkeley.edu

design for the deployed Internet of Things (IoT) sensors.

8.1.1 Description of the Social Game Experiment

Our experimental environment is comprised of residential housing single room apartments on the Nanyang Technological University campus. We designed a social game such that all single room dorm occupants could freely view their daily room's resource usage with a convenient interface. In each dorm room we installed two Internet of Things (IoT) sensors [3]—one close to the desk light and another near the ceiling fan. With the deployment of IoT sensors, dorm occupants can monitor in real-time their room's lighting system (desk and ceiling light usage) and HVAC (ceiling fan and A/C usage) with a refresh interval of up to one second.

Dorm occupants are rewarded with points based on how energy efficient their daily usage is in comparison to their past usage before the social game was deployed. The past usage data that serves as our baseline is gathered by monitoring occupant energy usage for approximately one month before the introduction of the game for each semester. Using this prior data, we calculated a weekday and weekend baseline for each of an occupant's resources. We accumulate data separately for weekdays and weekends so as to maintain fairness for occupants who have alternative schedules of occupancy (e.g. those who tend to stay at their dorm room over the weekends versus weekdays). We employ a lottery mechanism consisting of several gift cards awarded on a bi-weekly basis to incentivize occupants; occupants with more points are more likely to win the lottery. Earned points for each resource is given by

$$\hat{p}_i^d(b_i, u_i) = s_i \frac{b_i - u_i^d}{b_i} \tag{8.1}$$

where $\hat{p}_i^d$ is the points earned at day d for room's resource i which corresponds to ceiling light, desk light, ceiling fan, and A/C. Also, b_i is the baseline calculated for each resource i, u_i^d is the usage of the resource at day d, and s_i is a points booster for inflating the points as a process of framing [174]. This process of framing can greatly impact a user's participation, and it is routinely used in rewards programs for credit cards among many other point-based programs used in industry applications. In addition, we rewarded dorm occupants for the percentage of savings (8.1) because we felt it was important to motivate all of the participants to optimize their usage independent of the total amount of energy consumed in their normal schedule. However, over-consumption resulted in negative points.

In Figure 8.1, we present how our graphical user interface was capable of reporting to occupants the real-time status (on/off) of their devices, their accumulated daily usage, time left for achieving daily baseline, and the percentage of allowed baseline being used by hovering above their utilization bars. In order to boost participation, we introduced a randomly appearing coin close to the utilization bars with the purpose of incentivizing occupants to log in to web-portal and view their usage. The coin was designed to motivate

[3] *IoT sensor Tag*: http://www.ti.com/ww/en/wireless$_c$*onnectivity/sensortag/index.html*

occupants towards viewing their resource usage and understanding their impact to energy consumption by getting exact usage feedback in real-time. Based on this game principle, we gave occupants points when they clicked on the coin, which could increase both their perceived and actual chances of winning the rewards.

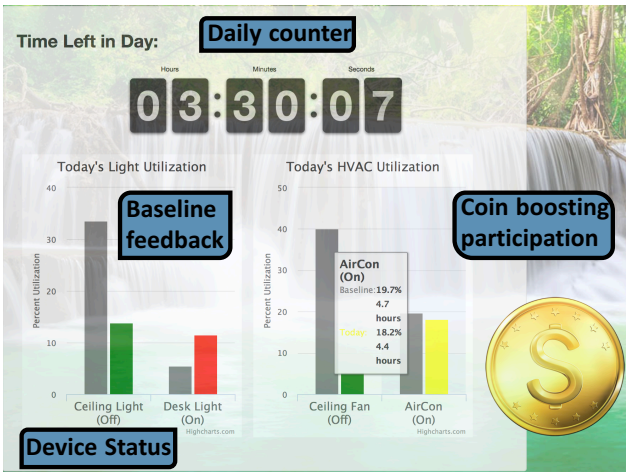
The residential housing single room apartments on the Nanyang Technological University campus are divided into four blocks, each of which having two levels. In this space, there is a total of seventy-two occupants who are eligible to participate in the social game. Participation in our social game platform was voluntary. We ran the experiment in both the Fall 2017 (September 12th - December 3rd) and Spring 2018 (February 19th - May 6th) semesters. In the Fall 2017 version, we included ceiling light, desk light, and ceiling fan resources in the graphical user interface for the social game while in the Spring 2018 version we included all of the potential resources that were available.

8.1.2 Internet of Things (IoT) System Architecture

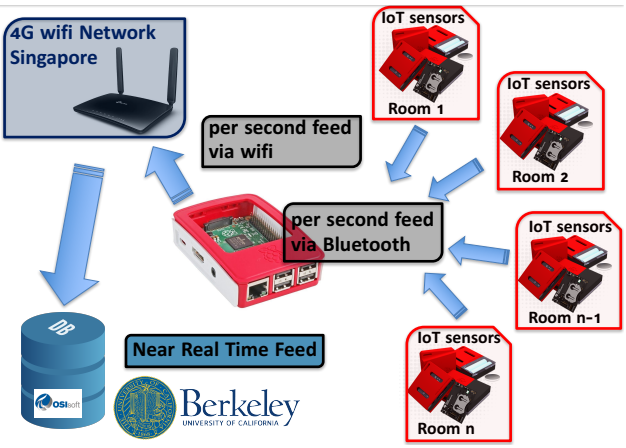
We enabled the design and implementation of a large-scale networked social game through the utilization of cutting-edge Internet of Things (IoT) sensors. In total, we have deployed one hundred and forty-four sensors in dorm single rooms. These are part of a hardware and software infrastructure that achieves a near real-time monitoring of several metrics of resource usage in each room like lighting and A/C, as well as recording detailed indoor conditions. Moreover, our system is capable of saving occupant actions in the web-portal as well as gathering weather data from our installed local weather monitoring station, which acts as an external parameter for our model. Weather data is gathered from an externally-installed local weather monitoring station at per-second resolution.

The actual design and dataflow is depicted in Figure 8.1. IoT sensors' software (firmware) has been updated for achieving highest reliability, continuous posting of indoor conditions in dorm rooms, and optimization of battery usage. In addition, we have designed and printed our own 3D plastic case for securing IoT sensors and for adding larger capacity battery (3V) for longer lasting capabilities. Our deployed IoT sensors are capable of continuously working for six months with a newly installed CR123 lithium battery. The IoT sensors continuously feed data every ten seconds via Bluetooth connection to a nearby deployed Raspberry Pi. To increase the system's reliability, we installed one Raspberry Pi in every other single dorm room. Next, each Raspberry Pi feeds the collected data via Wi-Fi to our deployed Wi-Fi routers connected to the Internet via 4G cellular modems. We have deployed our own 4G Wi-Fi Network since Nanyang Technological University residential housing doesn't have stable campus-wide Wi-Fi. In this way, our system can be deployed and operate independent of the condition of on-site utilities such as Wi-Fi. After receiving the Bluetooth feed, the data is posted to our OSIsoft PI database [4] located at the University of California, Berkeley campus.

[4] *OSIsoft PI database*: https://www.osisoft.com



(a) Graphical user interface (GUI) for energy-based social game: Display of consumption for resources, real-time feedback for device status (on/off), daily counter, and random coin



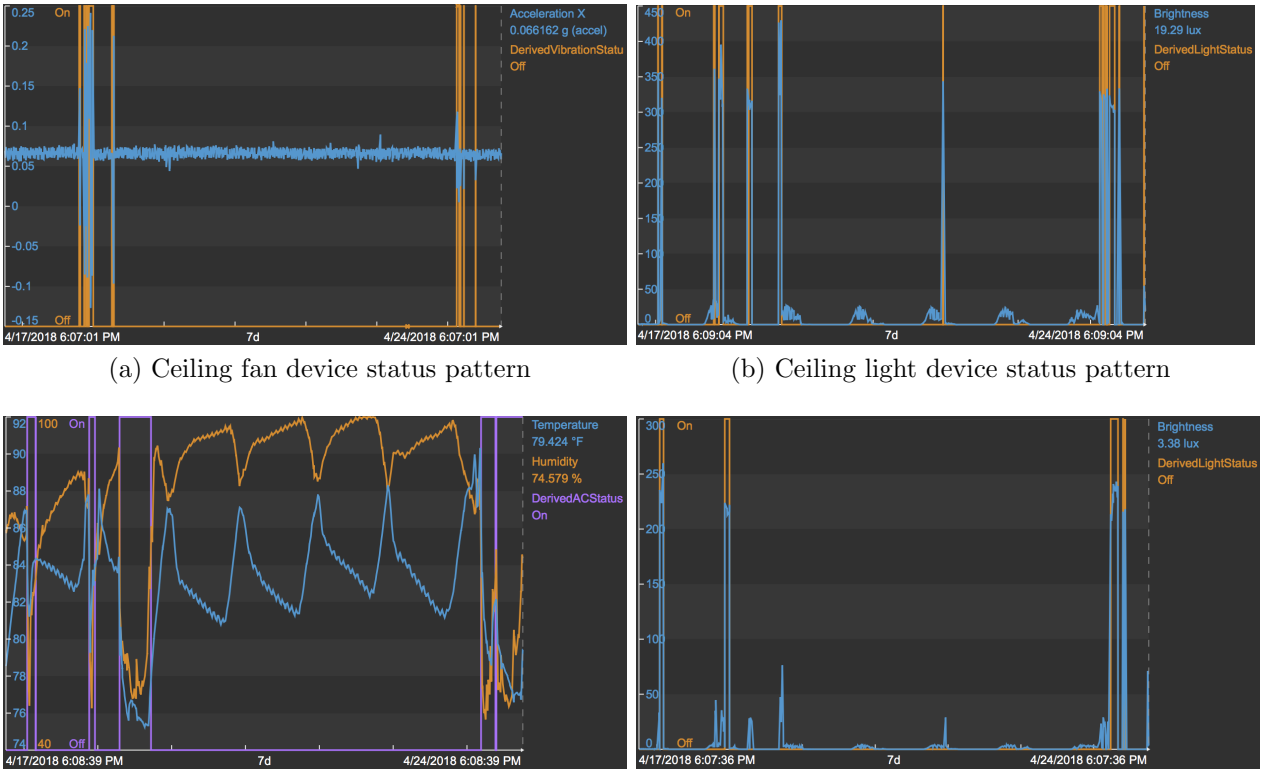
(b) Social game dataflow architecture design

Figure 8.1: Graphical user interface (GUI) and dataflow design for energy-based social game

Utilizing the data gathered from each dorm room, we leveraged several indoor metrics like indoor illuminance, humidity, temperature, and vibrations for the ceiling fan sensor. Having performed various tests during Summer 2017 within the actual unoccupied dorm rooms, we have derived simple thresholds indicating if a resource is in use or not. For instance, the standard deviation of acceleration derived from the ceiling fan mounted sensor is an easy way to determine whether ceiling fan is in the on state. Additionally, by combining humidity and temperature values, we were able to reliably identify whether A/C is in use with limited false positives. In Figure 8.2, we have included row (per second) data and their trends when indoor resources were on/off. Our calibrated detection thresholds are robust over daylight patterns, external humidity/temperature patterns, and noisy measurements naturally acquired from IoT sensors.

While we were getting streaming data from various sensors in all dorm rooms, our back-end processes updated the status of the devices in near real-time in each occupant's account and updated points based on their usage and point formula (8.1). This functionality allows occupants to receive feedback for their actions and view their points balance and their ranking among other capabilities. In order to allow participants to assess and visualize their energy efficient behavior, each user's background in the web-portal changes based on their ranking and energy efficiency. We used background pictures of rain forest settings for encouraging the more energy efficient occupants and images of desert scenes to motivate those with limited energy savings. For a live view of our web portal [5], you can visit our demo web-site, which serves as a demonstration of the game and as a hub for downloading de-identified per-

[5] *smartNTU* demo web-portal: https://smartntu.eecs.berkeley.edu



(a) Ceiling fan device status pattern

(b) Ceiling light device status pattern

(c) A/C device status pattern

(d) Desk light device status pattern

Figure 8.2: Patterns among all targeted resources. The on pulses point to instances that the thresholding system indicates activity in the device.

minute aggregated data. The entire web portal and background processes were developed using Django, a python based web development framework. MySQL was used as a database back-end for game related data not stored in the OSIsoft time series Pi database.

8.1.3 Social Game Data Det

As a final step, we aggregate occupants' data in per-minute resolution. We have several per-minute features like time-stamp, each resource's 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. In addition to these features, we add several external weather metrics like humidity, temperature, and solar radiation among others.

After gathering a high dimensional data set with all of the available features, we propose a **pooling & picking** scheme to enlarge the feature space and then apply a **Minimum Redundancy and Maximum Relevance (mRMR)** [134] feature selection procedure to identify useful features for our predictive algorithms. We pool more features utilizing a subset

of the already derived features by leveraging domain knowledge and external information. Specifically, we consider:

1. **College schedule dummy feature indicators:** including dummy variables for dates related to breaks, holidays, midterm period, and the final exam schedule at Nanyang Technological University. These features can capture occupants' variability of normal usage of their dorm room resources.

2. **Seasonal dummy feature indicators:** we compute several seasonal dummy variables utilizing our time-stamps. Examples of seasonal variables are time of day (morning vs. evening) and time of week (weekday vs. weekend) indicators. The intuition behind such features is their ability to capture occupants' seasonal patterns in resource usage.

3. **Resources status continuous features:** we incorporate stream resource status data for defining pooled features that accurately model occupants variability in resources usage across each day. Examples of such pooled features are: frequency of resource daily switches and percentage of resource usage across the day.

For more details related to our data set and feature space, visit our web site [6], which includes detailed descriptions of our features.

8.2 Human Decision-Making: Game Theoretic Framework

In this section and motivated by Section 2.2.2, we abstract the agents' decision-making processes in a game-theoretic framework under discrete choice theory. Under a discrete choice model, the possible outcome of an agent can be predicted from a given choice set using a variety of available features describing either external parameters or characteristics of the agent. We use a discrete choice model as a core abstraction for describing occupant actions related to their dorm room resources.

8.2.1 Agent decision-making Model

Consider an agent i and the decision-making choice set which is mutually exclusive and exhaustive. The decision-making choice set is indexed by the set $\mathcal{I} = \{\mathcal{J}^1, \ldots, \mathcal{J}^S\}$. Decision maker i chooses between S alternative choices and would earn a **representative utility** f_i for $i \in \mathcal{I}$. Each decision among decision-making choice set leads agents to get the highest possible utility, $f_i > f_j$ for all $i, j \in \mathcal{I}$. In our setting, an agent has a utility which depends on a number of features x_z for $z = 1, \ldots, N$. However, there are several unobserved components

[6] *smartNTU* demo web-portal: https://smartntu.eecs.berkeley.edu

— features of the representative utility which should be treated as random variables. Hence, we define a **random utility** decision-making model for each agent given by

$$\hat{f}_i(x) = g_i(\beta_i, x) + \epsilon_i \tag{8.2}$$

where ϵ_i is the unobserved random component of the agent's utility, $g_i(\beta_i, x)$ is a nonlinear generalization of agent i's utility function, and where

$$x = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N) \in \mathbb{R}^N \tag{8.3}$$

is the collective n features explaining an agent's decision process. The choice of nonlinear mapping g_i and x abstracts the agent's decision; it could represent, e.g., how much of a particular resource they choose to use and when an agent optimizes its usage over a specific resource. In general, each agent is modeled as a *utility maximizer* that seeks to select $i \in \mathcal{I}$ by optimizing (8.2).

Discrete choice models in their classical representation [173] are given by a linear mapping $g_i(\beta_i, x) = \beta_i^T x$ in which ϵ_i is an independently and identically distributed random value modeled using a Gumbel distribution. According to [173, Chapter 3], the probability that agent i chooses choice $j \in \mathcal{J}$ is given by

$$P_i^j = P(\beta_j^T x + \epsilon_j > \beta_z^T x + \epsilon_z, \forall j \neq z) \Longrightarrow P_i^j = \frac{\exp \beta_j^T x}{\sum_{s=1}^{\mathcal{J}} \exp \beta_z^T x} \tag{8.4}$$

According to (8.4), each agent's probability of a specific choice is given by a Logit model from the linearity assumption for the feature representative utility and the Gumbel distribution modeling the unknown random variable. Other distributions could be used (e.g. Gaussian for Probit model), and this is a design flexibility of discrete choice models.

8.2.2 Game Formulation

To model the outcome of the strategic interactions of agents in the deployed social game, we use a *sequential non-cooperative discrete game* concept. Introducing a generalized decision-making model for each agent (8.2), in which random utility can be modeled either with linear or nonlinear mapping, a sequential non-cooperative discrete game is given by

Definition 8. *Each agent i has a set $\mathcal{F}_i = f_i^1, \ldots, f_i^N$ of N **random utilities**. Each random utility j has a convex decision-making choice set $\mathcal{I}_j = \{\mathcal{J}_j^1, \ldots, \mathcal{J}_j^S\}$. Given a collective of n features (8.3) comprising the decision process given also the temporal parameter T, agent i faces the following optimization problem for their **aggregated random utilities**:*

$$\max\{\sum_{i=1}^{N} f_i^T(x) |\ f_i \in \mathcal{F}_i\}. \tag{8.5}$$

In the sequential equilibrium concept, we simulate the game defined by the estimated random utility functions per resource to demonstrate the actual decision-making process of each individual dorm occupant. Agents in the game independently co-optimize their aggregated random utilities (8.5) given a collective of n features (8.3) at each time instance. A general incentive design mechanism (8.1) motivates their potential actions across various given decision-making choice sets.

The above definition extends the definition of a discrete choice model [173] to sequential games in which agents concurrently co-optimize several discrete (usually mutually exclusive) choices. Using this definition, we can apply the proposed game theoretic model by allowing several machine learning algorithms to be directly applied. Machine learning algorithms can potentially be used for modeling the choice of nonlinear mapping (8.2). In particular, Deep Learning models can perform an end-to-end training for higher predictive accuracy using several mini-batched collectives of features (8.3).

8.3 Benchmark Learning Framework

In the previous section, we introduced an extension to discrete choice models for sequential decision-making over a set of different (usually mutually exclusive) choices. With appropriate modeling of unobserved random components and a linearity assumption for the features mapping in an agents utility function, we have a Logit model (8.4). However, unobserved random components and the overall structure of random utility (8.2) can be modeled by a variety of classification machine learning models—either discriminative types like the Logit model or even with generative models like Bayesian Networks.

In the current section, we examine the utility learning problem using a novel pipeline including a variety of proposed statistical learning methods and models that serve to improve estimation and prediction accuracy for our proposed sequential discrete choice model. The proposed benchmark learning framework scheme can be folded into an overall incentive design framework by either a building manager or utility companies. This goal motivates why we are interested in learning more than a simple predictive model for agents, but rather an exceptional utility-based forecasting framework that accounts for occupants' preferences. Furthermore, well-trained classification models serve as an excellent benchmark for our proposed Deep Learning models or other more advanced and complex sequential learning techniques like Hidden Markov Models or Conditional Random Fields.

8.3.1 Random Utility Estimation Pipeline

We start by describing the basic components of our proposed random utility estimation pipeline using observed pooled features and data derived from the game played between the agents. The utility learning framework we propose is quite broad in that it encompasses a wide class of discrete choice games, as our proposed game (8.5) is a super-set containing

classical discrete choice models. Let us introduce the pipeline formulation as it serves as the basis for the random utility estimation method.

After gathering streaming data in our MySQL data-base (as described in Section **??**), we pool several candidate features and expand our feature space. Next, a large set of proposed high dimensional candidate features is constructed. Using this feature set, we adopt a greedy feature selection algorithm called Minimum Redundancy Maximum Relevance (mRMR) [134]. The mRMR greedy heuristic algorithm utilizes mutual information as the metric of goodness for a candidate feature set. Given the large number of pooled candidate features, mRMR feature selection is a useful method of finding a subset of features that are relevant for the prediction of occupants' resource usage. The algorithm resolves the trade-off between relevancy and redundancy of the derived feature set by simultaneously reducing redundancy in the features and selecting those most relevant to occupants' actual actions over time. Our target is to avoid adding redundant features that do not boost prediction (e.g. classification) accuracy, but instead cause extra learning noise and increase computation time.

The mRMR feature selection algorithm is applied to batched gathered data from the first game period either in the Fall or Spring version of the Social Game. From the total number of available feature candidates, we decided to keep nearly half of them. Running the mRMR feature selection algorithm using the pooled features for the Fall semester — with the dorm room ceiling fan being the target resource — yields results that make logical sense based on the context of the features being examined. Unsurprisingly, ceiling fan percentage of usage is the most relevant and least redundant feature while external humidity seems to be the second most important feature influencing an occupant's ceiling fan usage. This leads us to the conclusion that mRMR is a suitable solution for obtaining the most relevant features in the proposed novel pipeline for utility learning.

After getting a number of top performing features as a result of the mRMR greedy algorithm, we apply a simple data pre-processing step with mean subtraction — subtracting the mean across each individual feature. Mean subtraction centers the cloud of data around the origin along every dimension. On top of mean subtraction, we normalize the data dimensions by dividing each dimension by its standard deviation in order to achieve nearly identical scale in the data dimensions. However, the training phase of the random utility estimation pipeline has one potentially significant challenge, which is the fact that data in almost every resource data set is heavily imbalanced (e.g. the number of resources with 'off' samples is on the order of 10-20 times more than those with 'on' samples). This is normal considering occupants' daily patterns of resource usage in buildings, but it poses a risk of having potentially poorly trained random utility estimation models.

For optimizing around highly imbalanced data sets, we adapt the Synthetic Minority Over-Sampling (SMOTE) [24] technique for providing balanced data sets for each resource and for boosting prediction (e.g. classification) accuracy. SMOTE over-samples a data set used in a classification problem by considering k nearest neighbors (in feature space) of the minority class given one current data point of this class. The SMOTE technique creates several artificial data points by randomly considering a vector along the resulting k neighbors

with respect to a current data point. Then, this randomly chosen vector is multiplied by a random number $s \in [0, 1]$ and added to the current data point. The outcome vector of this procedure results in newly synthesized data points. The SMOTE algorithm can be initialized by leveraging a pre-processing phase with Support Vector Machines as a grouping step.

After the SMOTE step, we train several classifiers to model (8.2) as a final step for the random utility estimation pipeline. These proposed machine learning algorithms do not require strong assumptions about the data process. Moreover, we propose a base model of logistic regression (8.4). In an effort to improve the base discrete choice model, we include penalized logistic regression (regLR) trying $l1$ norm protocol (Lasso) for the model training optimization procedure among other classical classification machine learning algorithms. We perform a randomized grid search for optimizing classifiers using the Area Under the Curve (AUC) metric [42] aiming to co-optimize TPR (sensitivity) and FPR (1-specificity).

We use the Area Under the Receiver Operating Characteristic (ROC) Curve as our performance metric. ROC curves describe the predictive behavior of a binary classifier by concurrently plotting the probability of true positive rate (i.e. correct classification of samples as positive) over false positive rate (i.e. the probability of falsely classifying samples as positive). Using the AUC metric from the ROC curve, we can quantify performance using a single metric to estimate the predictive accuracy of our proposed machine learning classification models as random utility estimators. For training the proposed machine learning algorithms, we used k-fold cross validation combined with the AUC metric to randomly split the data into training and validation sets in order to quantify the performance of each proposed machine learning model in the training phase. We applied 10-fold cross validation, and for each machine learning model we tuned the output model based on those parameters from a random grid search that achieved the best predictive performance on the cross validation set.

Each machine learning algorithm used in our benchmark pipeline and their respective hyper-parameters are described below

1. **Logistic Regression (LR):** Logistic regression is the base model for discrete choice models (8.4). Using a simple sigmoid function, it combines a set of linear features for achieving a posterior classification distribution.

2. **Penalized Logistic Regression (penLR):** Penalized logistic regression combines the cost function of classical Logistic regression with either $l2$ norm (Ridge) or $l1$ norm (Lasso) as a penalty term in the optimization procedure. Ridge and Lasso shrink or control the resulting weights in the obtained model. Lasso tends to result in more sparse models in which several weights can be zero or very close to zero [46]. Both Ridge and Lasso penalized logistic regression models are controlled by the λ hyper-parameter, which adjusts the penalty term in the optimization scheme.

3. **Bagged Logistic Regression (bagLR):** Bagging [46] is a powerful ensemble method for combining several weak classifiers and for building a prediction model in which majority vote scheme is applied. Bagging is a technique for reducing the overall variance

in the resulting machine learning model. It works based on a bootstrapping technique for re-sampling with replacement for N replicates of the original training data. Then we train N different logistic regression models and combine the resulting bootstrapped estimators by majority vote scheme. There is no immediate hyper-parameter for bagging other than selecting the number of models in which the bagged model will be constituted.

4. **Linear Discriminant Analysis (LDA):** The linear discriminant analysis classification algorithm can be considered as an alternative version of the base model for discrete choice models (8.4). The main difference lies in the fact that the LDA classifier infers joint probability distribution of the decision-making features and target (mutual exclusive) values. It defines a prior over the frequency of target values. It also fits a multivariable Gaussian distribution for the modeling of a conditional distribution given an individual target's values. Usually, it gives similar results to Logistic Regression (LR) despite its stronger modeling assumptions.

5. **k-Nearest Neighbors (kNN):** k-Nearest Neighbors classification algorithm is a non-parametric method which finds the k closest resulting training data points sampled across the whole of a given feature space [46]. The number of neighbors k is a hyper-parameter which largely controls this classifier's performance. For the k-Nearest Neighbors classifier case, the output given a testing sample is a class member obtained by a majority vote scheme against its neighbors. The k-Nearest Neighbors classification model is easy to build with a fast training phase. However, it has the requirement of caching training data points. Furthermore, it has a computationally expensive testing phase, which makes it impractical in real application settings.

6. **Support Vector Machine (SVM):** The support vector machine model obtains optimal hyperplane(s), which achieves a maximum separation margin between data classes [46]. The support vector machine model is controlled by the C hyper-parameter, which adjusts the separation of hyperplane(s) using maximum margin. Moreover, the support vector machine model could combine either $l2$ norm (Ridge) or $l1$ norm (Lasso) as an additional penalty term. The λ hyper-parameter again adjusts the penalty term in the optimization scheme.

7. **Random Forest:** Random Forest [46] is an efficient ensemble learning method used for classification among other machine learning tasks. It is a variation of classical bagging over decision trees. The Random Forest classifier constructs a large set of decision trees using classical CART greedy algorithm on a randomized subset of given input features. The randomization of input features allows significant reduction of variance in the generalization error. The output of the Random Forest classifier is given by the majority vote of the set of decision trees. One of the advantages of a Random Forest classifier is that testing a new data point takes $O(N * log(h))$ computation time, where N is the number of decision trees and h the height of each decision tree, which happens

to be highly balanced. Lastly, Random Forest has several hyper-parameters such as number of decision trees in the model, subset of features to include in the training phase, and depth of each binary trees among others. One significant note is the fact that a potentially large number of used decision trees somehow seem not to lead to over-fitting mainly due to randomization of feature set [46].

8.4 Leveraging Deep Learning for Sequential decision-making

Let us now formulate a novel Deep Learning framework for random utility estimation that allows us to drastically reduce our forecasting error by increasing model capacity and by structuring intelligent deep sequential classifiers. The architecture for deep networks is adaptive to proposed sequential non-cooperative discrete game models and achieves a tremendous increase to forecasting accuracy. Hence, deep networks achieve an end-to-end training for modeling agents' random utility (8.2) with extraordinary accuracy.

The primal version of Artificial Neural Networks (ANNs) can be dated back to 1943 [124]. After the emergence of Deep Learning, several versions of Artificial Neural Networks have solved a variety of challenging problems in areas such as recognition [56, 163, 169], classification in high-dimensional data sets [105], image caption [177, 185], machine translation [182], and generative models [150]. Due to ease of access to big data and the rapid development of adaptive artificial intelligence techniques, energy optimization and the implementation of smart cities has become a popular research trend in the energy domain. Researchers have deployed Deep Learning and Reinforcement Learning techniques in the field of energy prediction [40, 128] and intelligent building construction [121].

Deep Learning is a sub-field of machine learning that aims to extract multilevel features by creating a functional hierarchy in which higher level features are defined based on lower level features. The difference between the Deep Learning structure and the traditional single layer fully-connected neural network is that the use of more hidden layers in the network architecture between the input layer and the output layer of the model effectively achieves more complex and nonlinear relationships. In recent years, this concept has drawn strong interests as it has become the state-of-the-art solution to solve many practical problems in several regression [186], classification, as well as unsupervised learning problems [111, 140].

In our framework of random utility learning in a non-cooperative game setting, deep networks work as powerful models that can generalize our core model (8.2) by increasing capacity and by working towards an intelligent machine learning model for predicting agent behavior.

8.4.1 Deep Neural Networks for decision-making

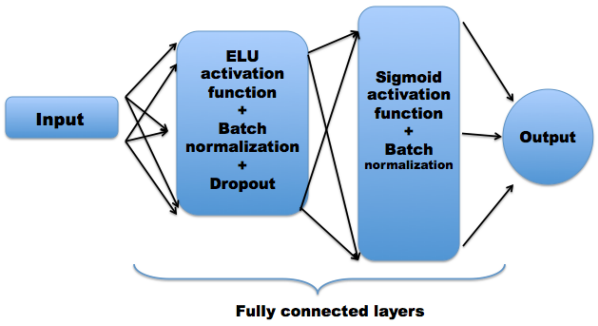
Deep neural network techniques have drawn ever-increasing research interests ever since Deep Learning in the context of rapid learning algorithms was proposed in 2006 [60]. Our

approach has the inherent capacity to overcome deficiencies of the classical methods that are dependent on the limited series of features located in the training data set (e.g. in our setting are the features resulting from mRMR feature selection algorithm). A deep neural network can be seen as a typical feed-forward network in which the input flows from the input layer to the output layer through a number of hidden layers (in general there are more than two). Our proposed deep neural network for random utility learning is depicted in Figure 8.3. The number of used layers can be dynamic and is a hyper-parameter that can be tuned. Usually two or three hidden layers are enough to represent a large capacity model. It is expected that a deep neural network model compared with a single hidden layer architecture achieves better performance in the classification predictive accuracy.

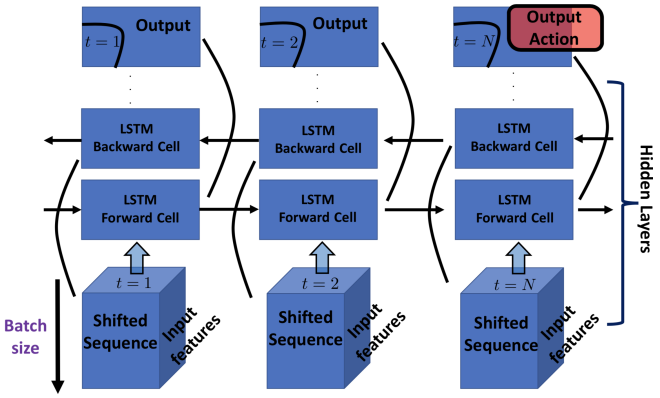
Our proposed deep neural network model for random utility learning includes exponential linear units (ELUs) [54] at each hidden layer. The usage of exponential linear units (ELUs) [54] normally adds an additional hyper-parameter in the search space as a trade-off for significant increases in fitting accuracy due to enormous decrements of "dead" units — a classical problem of rectified linear unit (ReLU) implementations [28]. The output layer is modeled using sigmoid units for classifying agents' discrete choices. The proposed model is optimized by minimizing the cross-entropy cost function using stochastic gradient descent combined with a Nesterov optimization scheme. The initialization of the weights utilizes He normalization [59] which gives increased performance and better training results. Unlike a random initialization, He initialization avoids local minimum points and makes convergence significantly faster. Batch Normalization [66] has also been adapted in our deep neural network framework to improve the training efficiency and to address the vanishing/exploding gradient problems in the training of deep neural networks. By using Batch Normalization, we avoid drastic changes in the distribution of each layer's inputs during training while the deep network's parameters of the previous layers keep changing. Knowing that adding more capacity in our deep neural network model will potential lead to over-fitting, we apply a dropout technique [167] as a regularization step. Using dropout, we apply a simple technique in the training phase (both in forward and backward graph learning traversal steps): each neuron, excluding the output neurons, has a probability to be totally ignored. The probability to ignore a neuron is another hyper-parameter of the algorithm and normally gets values between 50% - 70%.

8.4.2 Deep Bi-directional Recurrent Neural Networks for Sequential decision-making

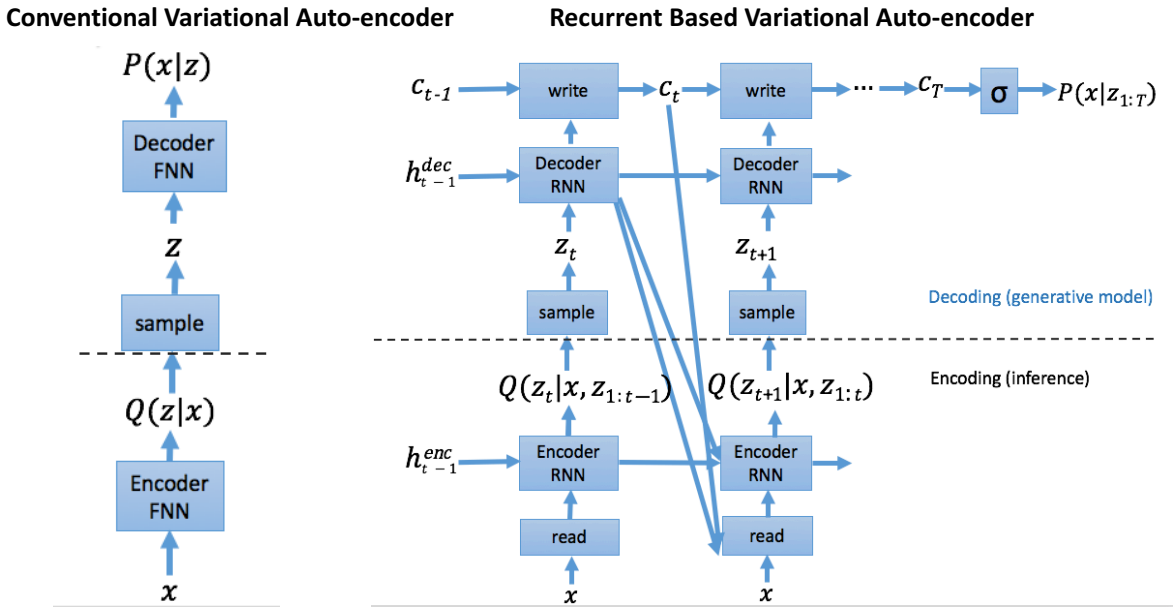
One of the basic drawbacks of both benchmark random utility learning models as well as the proposed deep neural networks is that they have strong assumptions for the data generation process. One important challenge for efficient learning of sequential decision-making models is the actual modeling of the dependence of future actions of an agent with the present and (or) previous action(s). In particular, an agent naturally tries to co-optimize around a set of discrete choices and gains the higher utility (8.5). Both benchmark models and deep neural



(a) Architecture of proposed deep neural network machine learning model for decision-making



(b) Sequential random utility estimation leveraging deep bi-directional recurrent neural networks



(c) Conventional and recurrent based deep auto-encoder

Figure 8.3: Proposed deep neural networks, deep bi-directional recurrent neural network and deep auto-encoders

networks adopt the assumption of **independent and identically distributed** data points. Hence, one way to model the underlying time series dependencies is through efficient feature engineering and by potentially using a novel feature selection algorithm. In Section 8.1.3, we use domain knowledge along with a pooling & picking method to create a feature set that can accurately predict agents' behavior. However, this step helps sparingly in the presence

of time series dependencies and cannot generalize.

Leveraging the latest Deep Learning models, like recurrent neural networks, we try mainly to address the issue of time dependence by looking at temporal dependencies within the data. Recurrent neural networks have the capability to allow information to persist, even over long periods, by simply inserting loops that point to them. As we see in the architecture of a deep bi-directional recurrent neural network in Figure 8.3, information passes from one time step of the network to the next. The information of the network passes to successor nodes. In the case of a bi-directional recurrent neural network, information flows also in the opposite direction to the predecessor. In a simple implementation however, recurrent neural networks tend to either vanish or become incapable of modeling long-term dependencies. In our proposed novel sequential utility learning model, we enable an end-to-end training using Long Short Term Memory cells (LSTM). LSTM cells overcome the problem of modeling long-term dependencies as they are designed explicitly for this reason. Equations (8.6), (8.7) describe exactly how LSTM cells work as an approach to modeling long term dependencies. Mainly, LSTM includes several gates that decide how long-term — short-term relations should be modeled. The overall output of the LSTM cell is a combination of sub-gates describing the term dependencies.

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \end{aligned} \tag{8.6}$$

$$\begin{aligned} j_t &= \sigma(W_{xj}x_t + W_{hj}h_{t-1} + b_j) \\ c_t &= f_t \otimes c_{t-1} + i_t \otimes j_t \\ h_t &= \tanh(c_t) \otimes o_t \end{aligned} \tag{8.7}$$

Our data pre-processing step as well as deep bi-directional architecture is described in Figure 8.3. Given an agent's actions, we define a time step N (sliding window of actions), which is a hyper-parameter and models time series dependencies in agent actions. Each training instance in the network is a tensor with the following dimensions:

- **mini-batch size:** Using mini-batch stochastic gradient descent, we select a batch size. Typically, we use some factors of time step N (such as 2-3 times N).
- **target sequence:** Similar to a normal time series input sequence for a deep neural network architecture, it has shifted tensors with appropriate time steps. It combines the tensor with a window of past actions (i.e. sliding window).
- **input features:** All available features from our data set.

Our deep network is designed to leverage sequential data and build several layers and time steps of hidden memory cells (LSTMs). Moreover, we propagate the unrolled deep network

both forward and backward (bi-directional recurrent neural network) for modeling the exact series time-lagged features for agents' actions. For the proposed deep bi-directional recurrent neural network, we use three hidden layers. To perform classification for the agents' actions, we add a fully connected layer containing two neurons (one per class). The fully connected layer is in turn connected to the output of the ending time-step propagating network, and this is finally followed by a soft-max layer, which actually performs the classification task. Similar to the deep neural network model, we optimize by minimizing the cross-entropy cost function using stochastic gradient descent combined with a Nesterov optimization scheme. Additionally, we employ an exponentially decaying learning rate as our learning rate schedule. Again, the initialization of the weights utilizes He normalization [59], which gives increased performance and better training results. For dealing with the enormous capacity of the proposed deep network, we apply dropout as a regularization step.

8.4.3 Deep Learning for Generative Sequential Decision Models

On top of the existing data set resulting from our experiment, researchers can create other even larger data sets based on the existing ones. The idea of bootstrapping [46] is widely applied both in statistics and machine learning in many applications along with the creation of new data sets mimicking the original one. However, bootstrapping is not a scalable solution, and as data sets become larger and larger, the computational complexity eliminates the capabilities of the system. In our approach, we propose the use of deep variational autoencoders [94] as an approach to create a nonlinear manifold (encoder) that can be used as a generative model. Variational auto-encoders formalize the necessary generative model in the framework of probabilistic graphical models by maximizing a lower bound on the log-likelihood of the given high dimensional data. Furthermore, variational auto-encoders can fit large high dimensional data sets (like our social game application) and train a deep model to generate data like the original data set. In a sense, generative models automate the natural features of a data set and then provide a scalable way to reproduce known data. This capability can be employed either in the utility learning framework for boosting estimation or as a general way to create simulations mimicking occupant behavior-preferences possible in software like EnergyPlus [7]

Using such a Deep Learning model, we can acquire generated samples by simply enabling the latent space of the auto-encoder and re-sampling using the decoder component. In Figure 8.3, we provide the overall idea behind training a variational auto-encoder, which resumes as a probabilistic auto-encoder. We use two hidden layers in encoder and decoder while tying parameters between them. Also, the latent space is modeled using a Gaussian distribution. By using this architecture of deep auto-encoder however, we limit the generative model in applications in which the data process has a natural time-series dependence. Hence, we proposed the implementation of a recurrent based variational auto-encoder [57]. In its architecture, shown in Figure 8.3, the proposed recurrent based variational auto-encoder

[7] https://energyplus.net

allows time-series modeling for progressive refinement and spatial attention in the shifted tensor inputs. Using progressive refinement, the deep network simply breaks up the joint distribution over and over again in several steps resulting in a chain of latent variables. This gives the capability to sequentially output the time-series data rather than compute them in a single shot. Moreover, a recurrent based variational auto-encoder can potentially improve the generative process over the spatial domain. By adding time-series in the model as tensors with shifted data points, we can reduce the burden of complexity by implementing improvements over small regions of the tensor input at a time instance (spatial attention).

Adapting those mechanisms, we achieve reduction of the complexity burden that an auto-encoder has to overcome. As a result, using a recurrent based variational auto-encoder allows for more generative capabilities that can handle larger, more complex distributions such as that in the given social game time-series. Our models were tested in several sets of data from individual occupants and was highly capable of randomly generating new data with extraordinary similarities to the training data. This fantastic result-adaptation provides an interesting tool for generating new data on top of the existing ones and provides more flexibility in the application of the data in several real scenario mechanisms like demand response.

8.5 Experimental Results

We now present the results of the proposed random utility learning method applied to high-dimensional IoT data collected from the network game experiment in the Fall 2017 and Spring 2018 semester. As we previously described, our data set consists of the per-minute high-dimensional data of occupants' usage across several resources in their rooms. We evaluate the performance of random utility learning under two characteristic scenarios: a) having full-information from the installed IoT sensors for performing Step-ahead predictions. In this scenario, IoT sensors are continuously feeding information from the previous actions of the occupants. b) under this scenario, called Sensor-free, we stop taking into account the IoT sensors' readings in each room. In the second case, the rich aggregated past-historic features of the occupants are missing. For this case, we have a model in which we use only features that we can acquire from external weather conditions (e.g. from a locally installed weather station), information about occupant engagement with the web-portal, and seasonal dummy variables. All of these features are much easier to be acquired without needing to keep the highly accurate but expensive IoT devices.

The broader purpose of our proposed gamification approach is the development of exceptional forecasting models representing occupants' dynamic behavior. As it is described in Figure 2.1 the acquired energy usage predictions can be fed to upper level components of smart grid as is provider or microgrid level. In a real application scenario, the proposed bottom-up modeling opens new avenues for demand respond programs, which incorporate real-time predictions of buildings—occupants energy patterns. The proposed game theoretic models and iterative incentive design mechanisms are powerful in a sense that can simulta-

neously used to predict but also to incentivize humans' behavior. Adaptive incentive design then motivates building occupants' energy efficiency through gamification platforms while accurately predict their energy usage—feed it back to upper provider levels.

We present estimation results for the complete data set in both Fall and Spring versions of the experiment for two characteristic occupants. The first occupant, in both Fall and Spring semester results, is considered a top rank player in the game with more aggressive behavior towards curtailing energy usage (e.g. in the results for this occupant in the Fall semester, there is not any usage of desk light in the testing data set—not even a minute). The second occupant, in both Fall and Spring semester results, is considered a middle rank player in the game with a behavior mixed with some energy efficient actions across several resources while also maintaining a daily usage pattern. We used the first four game periods for the training of our models:

- Fall semester training/testing data set: The training data set runs from September 12th, 2017 to November 19th, 2017 (ten weeks of data). It has approximately 100,800 distinct data points per occupant with per-minute resolution. The testing period considered the next bi-weekly game, from November 20th, 2017 to December 3rd, 2017, has a total of 20,160 distinct data points per occupant.

- Spring semester training/testing data set: The training data set runs from February 19th, 2018 to April 22nd, 2018 (nine weeks of data). It has approximately 90,720 distinct data points per occupant with per-minute resolution. The testing period considered the next bi-weekly game, from April 23rd, 2018 to May 6th, 2018, has a total of 20,160 distinct data points per occupant.

Before we trained our benchmark classifiers, we applied the mRMR algorithm to the total data set (all occupants data) in the training period. This accounts for almost 4 million distinct data points in the Fall semester data set and 2.5 million distinct data points in the Spring semester data set. mRMR results in several top features in both the Fall and Spring semester data sets. Interestingly, mRMR included several external features in the top relevant feature candidates. In particular, the presence of external humidity as an important feature for the ceiling fan is a good demonstration of the mRMR algorithm's capability to extract salient features. Moreover, features like survey points illustrate that some occupants co-optimized their resource usage while also trying to view their point balance, usage, and ranking in the game (e.g. visiting the web-portal).

8.5.1 Forecasting via Benchmark & Deep Learning Framework

We have dual objectives by leveraging benchmark & Deep Learning frameworks. Our first objective is to accurately forecast building occupants' resources usage, even in per minute resolution. Specifically, providers/retailers in a smart grid setting can integrate energy usage forecasts in demand response programs. Then, our second objective is to improve building

energy efficiency by creating an adaptive model that learns how user's preferences change over time and thus, generate the appropriate incentives to ensure active participation. Furthermore, the learned preferences can be adjusted through incentive mechanisms [145] to enact improved energy efficiency (seen in Figure 2.1).

For learning optimal random utility models in the benchmark setting, we use the top twenty-five resulting features from the mRMR algorithm along with a pre-processing step of SMOTE with SVM initialization. Using SMOTE, we boost the accuracy of benchmark models due to the fact that our data set was heavily imbalanced. We achieve decent accuracy using well-trained benchmark models. Area under the receiver operating characteristic (ROC) curve (AUC score) is our forecasting performance metric. For clarity, AUC score quantifies the predictive behavior of a binary classifier by concurrently plotting the probability of true positive rate (i.e. correct classification of samples as positive) over false positive rate (i.e. the probability of falsely classifying samples as positive). Especially in the Step-ahead predictions, all of the classifiers achieve decent AUC scores in both the Fall and Spring semester results, as shown in Tables 8.1, 8.3. In the Sensor-free results, Tables 8.2, 8.4, we have a significant drop in the achieved accuracy, but this is expected given that the IoT feed is decoupled. However, even in Sensor-free examples we are able to predict occupants' behavior using less representative features and having excluded the "costly" IoT sensors.

For the Deep Learning models, we used the exact same data set. For the deep neural networks, we used training data resulting from the applied SMOTE step as in the benchmark analogy. We used two hidden layers of the feed-forward neural network, with 50% dropout and stochastic gradient descent method leveraging Nesterov's Momentum to accelerate convergence.

To further exploit the continuity of the sequential decision-making model, we experiment on the bi-directional deep recurrent neural network. We used a time sliding window—time step of two hours (120 distinct data-points). We processed the data without being pre-processed from the SMOTE algorithm as we wanted to keep control of the underlying sequence of actions of the occupants (temporal dependences of the data). We used three hidden layers with 60% dropout rate and we applied exponentially decaying learning rate as our learning rate schedule. In the training of bi-directional recurrent neural networks, we applied the principle of early stopping using a validation data set (small portion of the data — one week period) over the AUC metric. For our deep bi-directional networks, thirty-five epochs were optimal to be trained.

To evaluate the effectiveness of our proposed deep learning framework, we present the AUC scores of a representative example for comparison. From the results, it is clear that deep RNN outperforms the majority of alternative algorithms with few outliers (bold highlighted scores). One important remark is that deep RNN performs best even when compared to Random Forest, which is considered a top robust performing classification model. Interestingly, deep NN could achieve better performance in some examples over the Random Forest classifier, but this is not a general case. Figures 8.4 and 8.5 introduces bar charts representing AUC scores for ceiling fan usage (On/Off). Step-ahead & Sensor-free AUC scores show the prediction of occupants' behavior. Clearly, deep RNN outperform all other

Deep Learning and machine learning models. Also, it is clear in Figures 8.4 and 8.5 that Deep Bi-directional RNN based models achieve accuracy almost equal to 1.

Occupant 1 / 2	Ceiling Fan	Ceiling Light	Desk Light
Logistic regression	0.74 / 0.83	0.75 / 0.78	N/A / 0.78
Penalized $l1$ Logistic regression	0.75 / 0.80	0.75 / 0.77	N/A / 0.78
Bagged Logistic regression	0.76 / 0.84	0.77 / 0.80	N/A / 0.79
LDA	0.75 / 0.81	0.75 / 0.78	N/A / 0.74
K-NN	0.70 / 0.76	0.72 / 0.77	N/A / 0.74
Support Vector Machine	0.74 / 0.82	0.76 / 0.78	N/A / 0.76
Random Forest	0.92 / 0.91	0.79 / 0.78	N/A / 0.98
Deep Neural Network	0.82 / 0.80	0.80 / 0.76	N/A / 0.78
Deep Bi-directional RNN	**0.93** / **0.97**	**0.89** / **0.85**	N/A / **0.99**

Table 8.1: AUC scores using Fall semester data of two representative occupants towards Step-ahead predictions.

Occupant 1 / 2	Ceiling Fan	Ceiling Light	Desk Light
Logistic regression	0.62 / 0.65	0.61 / 0.61	N/A / 0.68
Penalized $l1$ Logistic regression	0.59 / 0.65	0.55 / 0.56	N/A / 0.64
Bagged Logistic regression	0.64 / 0.66	0.61 / 0.59	N/A / 0.68
LDA	0.63 / 0.65	0.60 / 0.58	N/A / 0.68
K-NN	0.56 / 0.53	0.50 / 0.56	N/A / 0.55
Support Vector Machine	0.64 / 0.65	0.60 / 0.60	N/A / 0.68
Random Forest	0.58 / 0.60	0.56 / 0.59	N/A / 0.63
Deep Neural Network	0.59 / 0.55	0.53 / 0.60	N/A / 0.64
Deep Bi-directional RNN	**0.69** / **0.71**	**0.65** / **0.66**	N/A / **0.76**

Table 8.2: AUC scores using Fall semester data of two representative occupants towards Sensor-free predictions.

8.5.2 Generative Models via Sequential Deep Auto-encoders

In Table 8.5, we present the results of two trained generative models using the full data set of a randomly selected occupant in the Fall semester. We trained both a conventional auto-encoder and a recurrent based auto-encoder. The derived Deep generative models can be used as a way to create simulations for mimicking building occupants' behavior—preferences. This is an extra tool for quantifying variations in building occupants behavior. Moreover, generative models are capable of adapting variations of the external weather conditions, which in turn creates an interesting view of building occupants' energy usage patterns aligned with external weather patterns.

Occupant 1 / 2	Ceiling Fan	Air-con	Ceiling Light	Desk Light
Logistic regression	0.71 / 0.84	0.76 / 0.82	0.75 / 0.83	0.76 / 0.73
Penalized $l1$ Logistic regression	0.71 / 0.84	0.76 / 0.82	0.75 / 0.83	0.76 / 0.71
Bagged Logistic regression	0.73 / 0.85	0.73 / 0.83	0.76 / 0.84	0.79 / 0.74
LDA	0.70 / 0.87	0.73 / 0.83	0.75 / 0.83	0.70 / 0.92
K-NN	0.70 / 0.84	0.76 / 0.86	0.68 / 0.81	0.73 / 0.76
Support Vector Machine	0.70 / 0.86	0.75 / 0.83	0.75 / 0.83	0.70 / 0.49
Random Forest	0.83 / **0.99**	0.83 / 0.81	**0.99** / **0.98**	0.96 / 0.87
Deep Neural Network	0.74 / 0.86	0.78 / 0.87	0.77 / 0.84	0.84 / 0.90
Deep Bi-directional RNN	**0.91** / 0.98	**0.89** / **0.94**	**0.99** / 0.97	**0.99** / **0.95**

Table 8.3: AUC scores using Spring semester data of two representative occupants towards Step-ahead predictions.

Occupant 1 / 2	Ceiling Fan	Air-con	Ceiling Light	Desk Light
Logistic regression	0.55 / 0.71	0.73 / 0.61	0.55 / 0.69	0.50 / 0.70
Penalized $l1$ Logistic regression	0.55 / 0.72	0.70 / 0.61	0.55 / 0.70	0.50 / 0.73
Bagged Logistic regression	0.54 / 0.73	0.73 / 0.73	0.54 / 0.71	0.51 / 0.65
LDA	0.55 / 0.71	0.73 / 0.68	0.55 / 0.70	0.51 / 0.59
K-NN	0.50 / 0.63	0.57 / 0.69	0.54 / 0.69	0.57 / 0.50
Support Vector Machine	0.55 / 0.70	0.73 / 0.67	0.55 / 0.70	0.50 / 0.71
Random Forest	0.58 / 0.66	0.65 / 0.54	0.54 / 0.68	0.50 / 0.50
Deep Neural Network	0.56 / 0.67	0.68 / 0.53	0.54 / 0.67	0.50 / 0.50
Deep Bi-directional RNN	**0.66** / **0.79**	**0.80** / **0.78**	**0.64** / **0.77**	**0.62** / **0.83**

Table 8.4: AUC scores using Spring semester data of two representative occupants towards Sensor-free predictions.

In Table 8.5, we present the results of two trained generative models using the full data set of a randomly selected occupant in the Fall semester. We trained both a conventional auto-encoder and a recurrent based auto-encoder. In Table 8.5, we present several selected features, either from the interior of a dorm room or external weather data. For the evaluation of the artificially generated time-series using the proposed auto-encoders, we utilize dynamic time warping (DTW) for measuring the similarity between the two temporal sequences — the ground truth and the artificial data from the generative model. Dynamic time warping (DTW) is an extension of Levenshtein distance and can be computed in pseudo-polynomial time [158]. In bold, we see that the recurrent based auto-encoder achieves a smaller DTW score in most of the features, leading to a generative model that isn't mimicking exactly or is way too different from the original data set. Wanting to evaluate the statistical significance of the calculated DTW scores from the recurrent based auto-encoder, we used a permutation hypothesis test. In this approach, we permute original and generated time-series and we

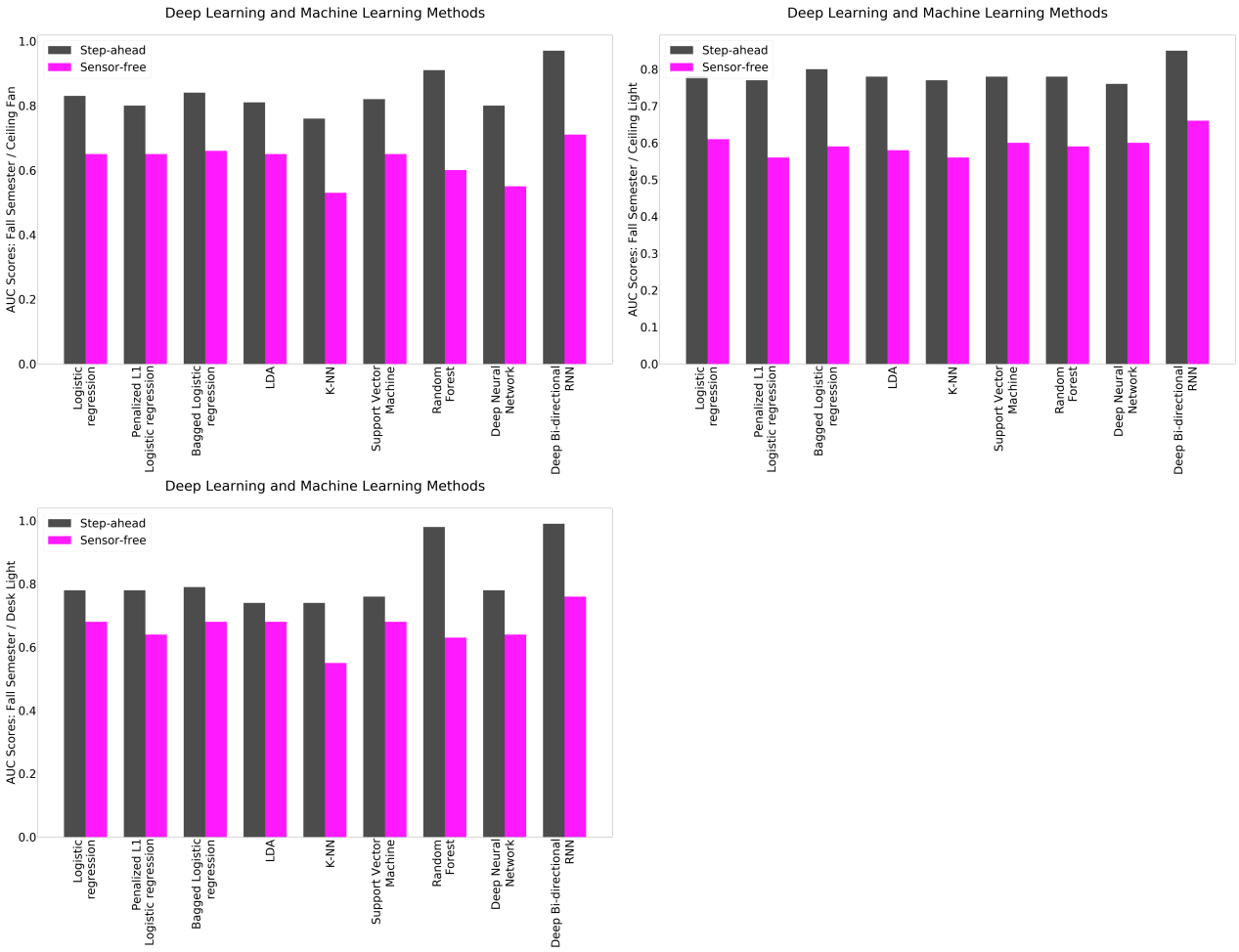


Figure 8.4: Forecasting accuracy (Step-ahead / Sensor-free predictions) for Fall semester data—resources (On/Off).

computed their DTW score looking for events that are more "extreme" than the one that is presented in Table 8.5. Interestingly, we have inside and outside weather based features (temperature and humidity) that have zero p-values showing that the DTW score using a recurrent based auto-encoder are significant. For indoor device status features however, p-values are large, showing that the DTW score has high variability under the permutation test.

8.5.3 Energy Savings through Gamification

Here we present achieved savings in both Fall and Spring semester version of the social game. Our gamification framework enables occupants to a friendly non-cooperative game and highly motivates the reduction of their energy usage. Through the deployed IoT sensors and developed web portal, each individual building occupant got live feedback about room's

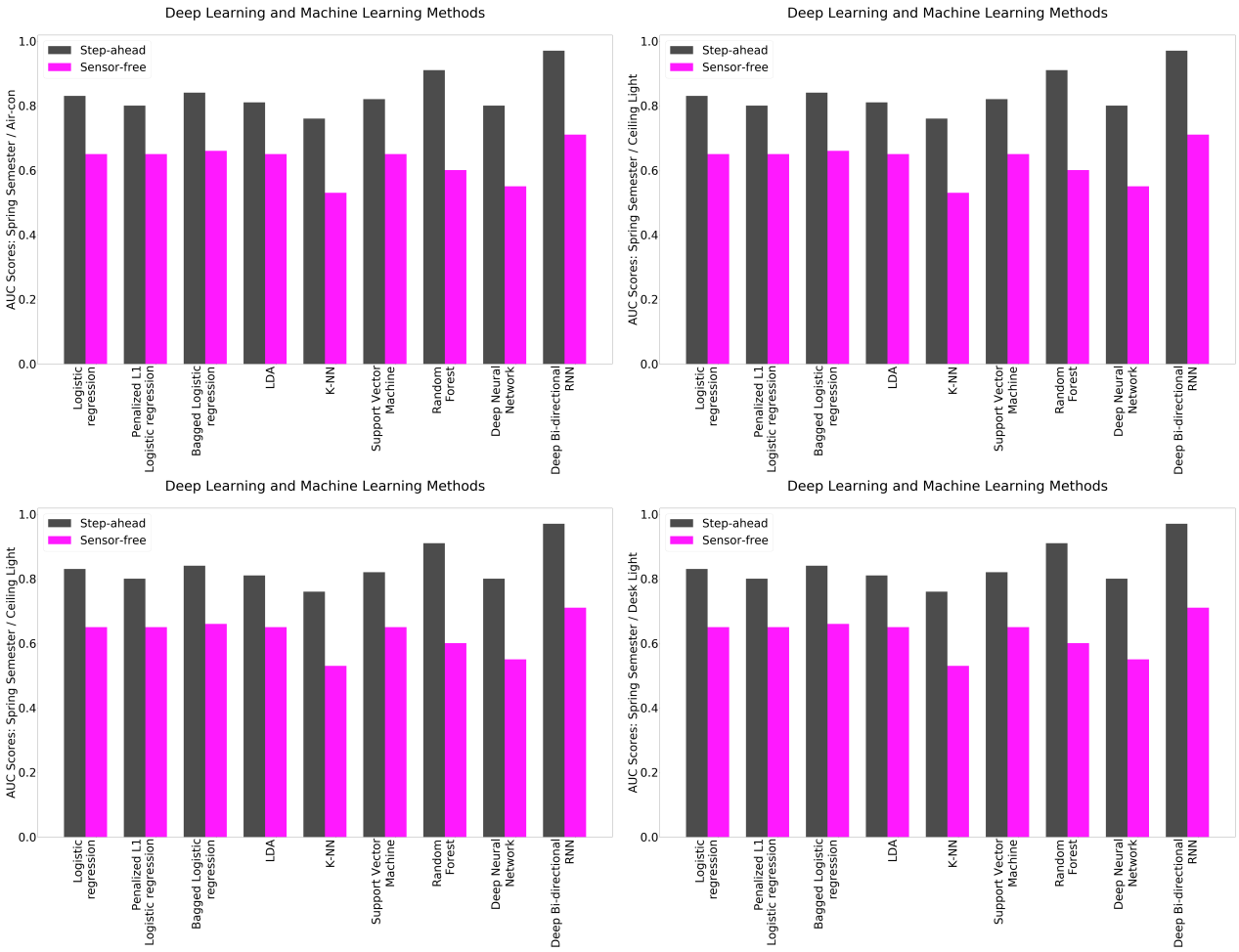


Figure 8.5: Forecasting accuracy (Step-ahead / Sensor-free predictions) for Spring semester data—resources (On/Off).

usage and energy efficiency throughout the day. In Figures 8.6 and 8.7 we present the daily average minutes usage compared to weekday & weekend average baselines.[8] The vertical black dashed lines indicate a weekend period, which has a different average baseline target for the occupants. In Table 8.6 we compare the occupants usage patterns, weekday vs weekend, after the beginning of the game. In particular, we perform 2-sample t-tests, which in most of the cases show that there is a significant difference in occupants' usage patterns between weekdays and weekends. These are significant results showing how we can optimally incentivize occupants in residential buildings to reduce energy usage, especially over weekend periods. This result verifies occupants' significant different usage patterns in weekdays &

[8]Weekday & weekend average baselines are computed using past usage data over a period of four weeks before the beginning of the social game.

Time Series Feature	Conventional	RNN-based	p-values
Ceiling Fan Status (On / Off)	1.5e+04	**1.2e+04**	0.11
Ceiling Light Status (On / Off)	1.6e+04	**2.2e+03**	1.0
Desk Light Status (On / Off)	6.7e+03	**0.0e+00**	1.0
Dorm Room Temperature	1.3e+05	**1.2e+05**	0.0
Dorm Room Humidity	4.8e+05	**3.7e+05**	0.0
External Temperature	1.0e+05	1.8e+05	0.0
External Humidity	2.9e+05	4.3e+05	0.0

Table 8.5: DTW score — feature comparison between proposed generative models (autoencoders).

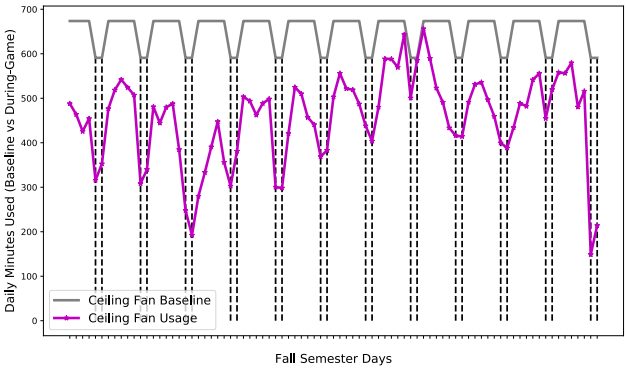
	Spring			Fall		
Device	Wday	Wkend	p-value	Wday	Wkend	p-value
Ceiling Light	314.2	195.6	0	393.9	257.5	0
Desk Light	104.6	81.5	0.1	157.5	123.3	0.01
Ceiling Fan	541.6	331.8	0	537.6	407.0	0
Air Con	225.8	81.9	1.0	N/A	N/A	N/A

Table 8.6: Weekday vs. Weekend Mean usage hypothesis testing

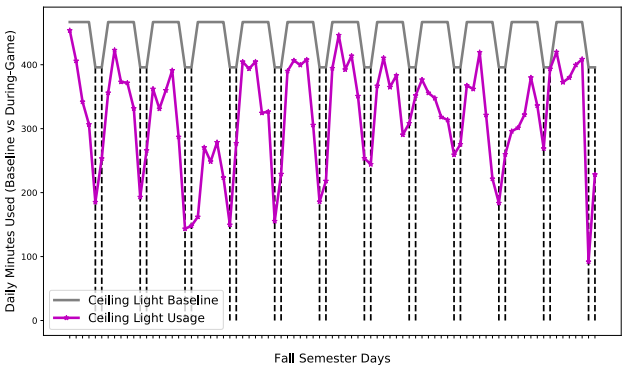
weekends. Regarding energy usage—savings, in Figures 8.6 and 8.7 desk light usage has the most significant reduction in usage compare to other sources. Moreover, all the usage sources have impressive reduction in usage.

For quantifying the results, we employ hypothesis testing (A/B testing) using dorm occupants' usage data before and after the beginning of the experiment. Hypothesis testing is a standard technique of great importance used across all fields of research. In the energy domain, we see many examples where testing is crucial in determining the feasibility of the hypothesis. As an example, testing energy-GDP causality has resulted in a lot of disparate results, largely because of omitted variable bias and the lack of a priori hypothesis testing [116].

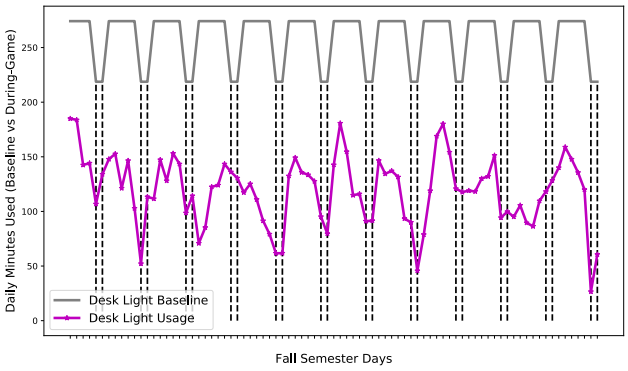
In Ang's paper, we see that hypothesis testing is central in determining the correlation between CO2 emissions and energy consumption in France [4]. Multiple tests are performed to find causal links of output energy and pollution. This is also complemented with Granger non-causality tests and exogeneity tests, for comparison between the short run and the long run. This is an idea that we could incorporate, since one of our goals is to provide a suitable long run forecast. Then, we would have a more informed idea of what we can expect in different time periods. This suggests a natural extension to our work in this paper, where we can further analyze the sequential discrete game to account for time-series dependencies. Additionally, hypothesis testing is frequently supplemented with other statistical procedures such as cross-validation and information criteria. This is relevant, since in this paper, we



(a) Ceiling fan daily average usage vs baselines



(b) Ceiling light daily average usage vs baselines



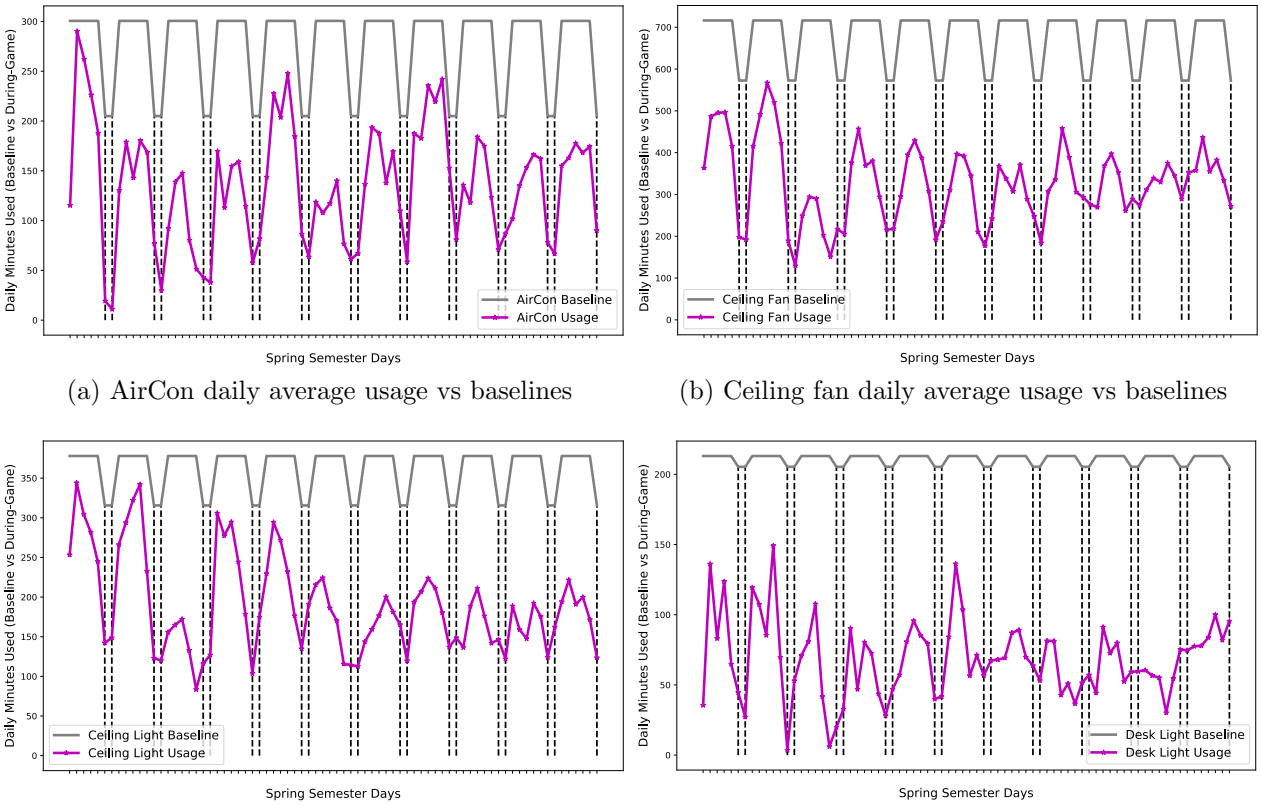
(c) Desk light daily average usage vs baselines

Figure 8.6: Fall semester daily average minutes usage compared to weekday & weekend average baselines. Vertical black dashed lines indicate a weekend period.

have attempted to predict and improve forecasting performance using various Deep Learning techniques.

In Tables 8.7 and 8.8, we see the hypothesis testing values for the different devices in both iterations of the experiment (Fall and Spring). In the tables, the Before column denotes the data points gathered from before the game was officially started. The After column is the data during the game period. Data points in the tables are bucketed in both weekday and weekend data and represent the average usage of all the occupants. Usage is defined in minutes per day. In all cases of the devices, we have a significant drop in usage between the two periods. Drop in usage is given in the column named Δ %, and indicates reduction in the average usage of all the participating occupants. The p-values resulting from the 2-sample t-tests show that the change in usage patterns is highly significant. Moreover, we can see that a much larger drop in usage is achieved over the weekends.

8.5.4 Survey Results



(a) AirCon daily average usage vs baselines

(b) Ceiling fan daily average usage vs baselines

(c) Ceiling light daily average usage vs baselines

(d) Desk light daily average usage vs baselines

Figure 8.7: Spring semester daily average minutes usage compared to weekday & weekend average baselines. Vertical black dashed lines indicate a weekend period.

	Weekday				Weekend			
Device	Before	After	p-value	Δ %	Before	After	p-value	Δ %
Ceiling Light	417.5	393.9	0.02	5.6	412.3	257.5	0	37.6
Desk Light	402.2	157.5	0	60.8	517.6	123.3	0	76.2
Ceiling Fan	663.5	537.6	0	19.0	847.1	407.0	0	51.9

Table 8.7: Fall Game (Before vs After) Mean usage hypothesis testing.

In this last section, we present the results of the given surveys in the Spring semester. In Figure 8.8 we present various demographic data from the participants in Spring semester. Figure 8.9 presents occupants' feedback about given baselines' difficulty—hardness. Baselines are calculated for each individual occupant based on previous historic data from each occupant's dorm room. However, Figure 8.9 gives a great feedback about how achievable are the given baselines. For example, desk light baselines seem to be an easy task for dorm occupants. This is a valuable information for building manager targeting for energy efficiency in

	Weekday				Weekend			
Device	Before	After	p-value	Δ %	Before	After	p-value	Δ %
Ceiling Light	452.0	314.2	0	30.5	426.0	195.6	0	54.1
Desk Light	430.1	104.6	0	75.7	509.4	81.5	0	84
Ceiling Fan	777.4	541.6	0	30.3	847.1	331.8	0	60.8
Air Con	469.8	225.8	0	51.9	412.3	81.8	0	80.2

Table 8.8: Spring Game (Before vs After) Mean usage hypothesis testing.

smart buildings. These type of questions can as prior knowledge for the gamification design as well as for inferring occupants' utility functions. In Figure 8.10 we present occupants answer regarding their consciousness about potential energy efficient actions while they were participating in the deployed social game. Also, occupants responded back why they did (did not) take energy efficient actions. Not surprisingly, the answer **"I am too busy"** was the most frequent whenever an occupant is not motivated to reduce energy usage in the dorm.

Surveys were administered to occupants every two days and included several questions using a 5-point Likert-scale survey format. For each question's target set, we phrased it in opposite terms and we randomly sorted the questions order at each instance of survey administration. Wanting to test occupants' internal consistency regarding survey results, we deployed a Cronbach's α statistic test. As we see in Table 8.9, people are internally consistent regarding their satisfaction for the lighting & HVAC condition and provided incentives. We also note that the Cronbach's α is negative for the 3rd and 4th entries, regarding the awareness of energy-saving techniques and the web portal interface. This is because there are weak correlations between the variables, which is counter intuitive seeing that the questions were designed such that consistent answers were encouraged. This result suggests that dorm occupants felt that the two questions in each bucket for the third and fourth categories were not actually asking the same question in essence.

8.6 Chapter Summary

We presented a general framework for random utility learning in sequential decision-making models. We leveraged several Deep Learning architectures and proposed a novel sequential Deep Learning classifier model, which utilizes bi-directional recurrent networks along with LSTM cells. We also introduced a framework that serves as a base for creating generative models ideal for modeling human-centric architectures. On top of that, we developed a detailed random utility pipeline for several classical benchmark models. The latter is important for having a variety of models to characterize occupants' behavior, but also to compare our proposed Deep Learning models.

To demonstrate the random utility learning methods, we applied them to data collected from a smart building social game we conducted where occupants optimized concurrently their room's resources and participated in a lottery. We were able to estimate several agent

	Questions	Cronbach's α	p-value
Light	I am satisfied with today's lighting conditions. Today's lighting conditions were uncomfortable.	0.75	0.003
Incentives	I am happy with the current incentives provided. The current incentives are not satisfactory.	0.82	0.002
Energy	I am aware of energy-saving techniques that I can use. I feel unable to save energy through my actions	-0.40	0
Web Portal	I am satisfied with the current web interface. The web portal leaves much to be desired.	-0.24	0
HVAC	I am satisfied with today's HVAC (thermo comfort aircon/ fan) conditions. Today's HVAC (thermo comfort – aircon/fan) conditions were uncomfortable.	0.78	0.003

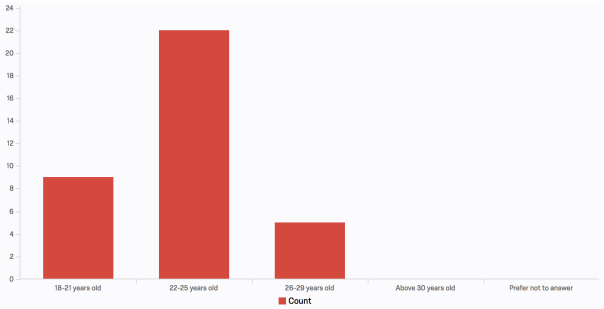
Table 8.9: Cronbach's α Testing for 5-point Likert-scale Survey Responses

profiles and significantly reduce the forecasting error compared to all benchmark models. The deep sequential random utility learning framework outperformed all the models, and, in specific examples, it improved prediction accuracy to an extraordinary degree. This last result shows that a Deep Learning architecture that handles a sequential data process boosts the overall accuracy. Although we apply the method specifically to smart building social game data, it can be applied more generally to scenarios with the task of inverse modeling of competitive agents, and it provides a useful tool for many smart infrastructure applications where learning decision-making behavior is crucial.

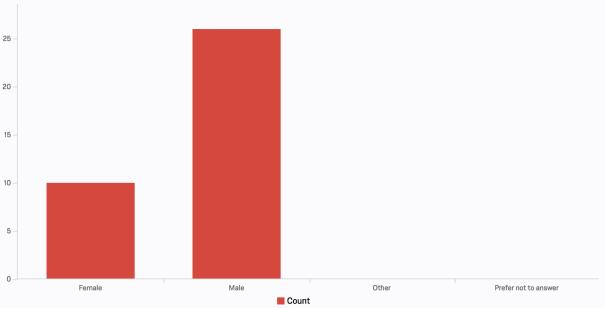
By running this gaming experiment on the Nanyang Technological University campus in both the Fall 2017 (September 12th - December 3rd) and Spring 2018 (February 19th - May 6th) semesters, we enabled the gathering of a large data set including not only occupants' gaming actions but also their resource usage, occupancy preferences, and interactions with web portal. After analyzing and formulating this data set, we designed a demo web portal for the demonstration of our infrastructure and for downloading de-identified high dimensional data sets[9]. Our provided high-dimensional data set can serve either as a benchmark for discrete choice model learning schemes or as a great source for analyzing occupant resource

[9] *smartNTU* demo web-portal: https://smartntu.eecs.berkeley.edu

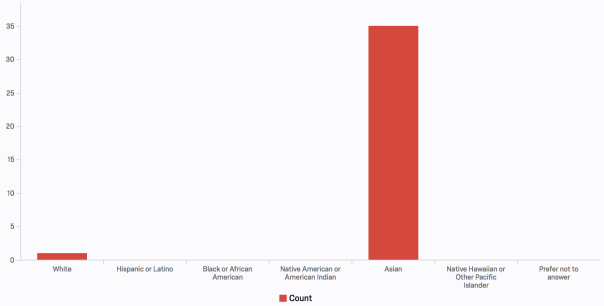
usage in residential buildings. Other researchers now have the ability to easily demonstrate gaming data in a discrete choice setting, run simulations including occupants dynamic preferences (data set has one minute resolution), test correlations of actions vs external parameters like weather (e.g. we provide various weather metrics), and leverage temporal data sets in several demand response program approaches [81, 82].



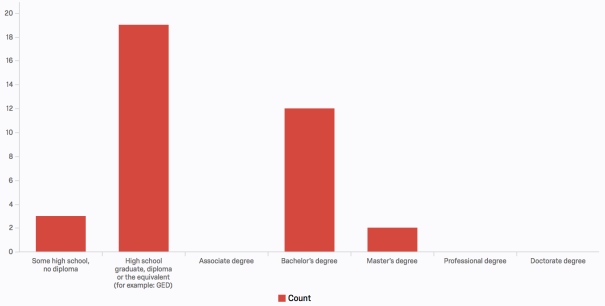
(a) Demographic data: Age range



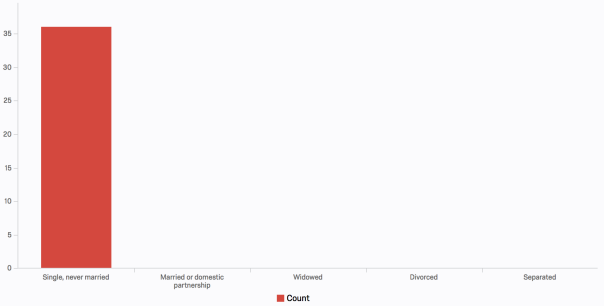
(b) Demographic data: Gender



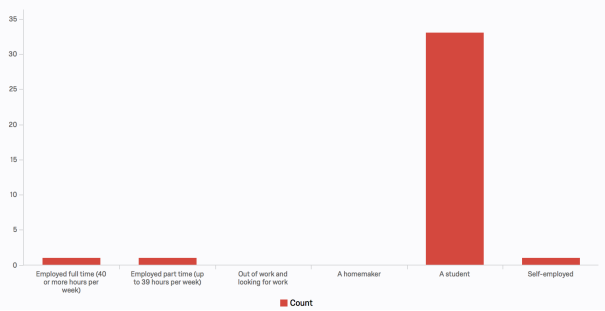
(c) Demographic data: Race



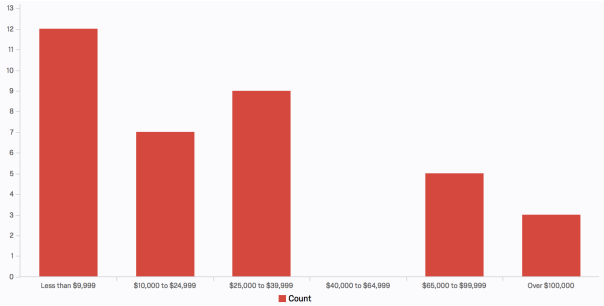
(d) Demographic data: Higher education background



(e) Demographic data: Marital status

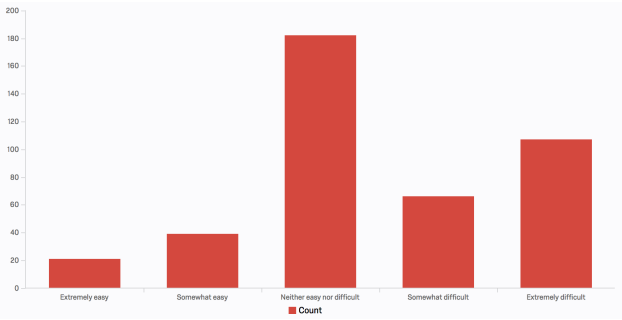


(f) Demographic data: Employment status

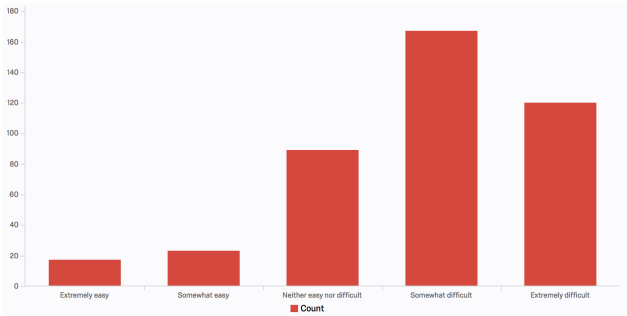


(g) Demographic data: Income range

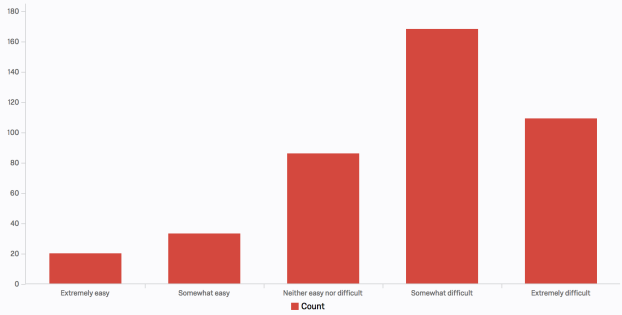
Figure 8.8: Various demographic survey results from Spring semester occupants



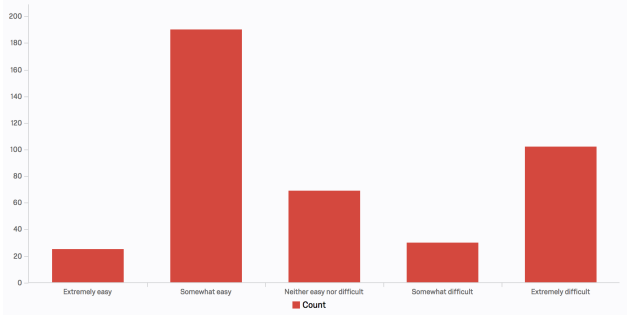
(a) How hard do you think your AirCon resource baseline can be achieved?



(b) How hard do you think your Ceiling Fan resource baseline can be achieved?

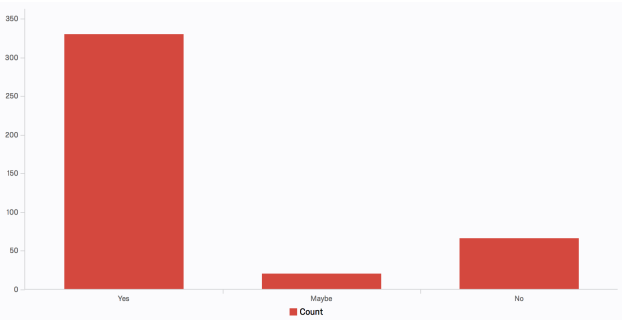


(c) How hard do you think your Ceiling Light resource baseline can be achieved?

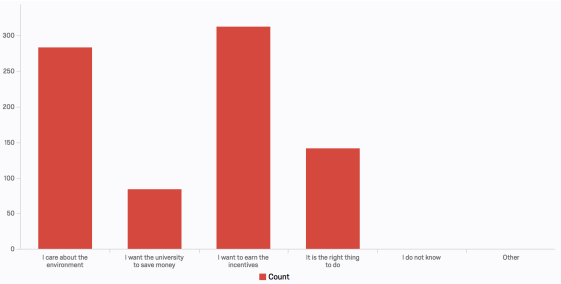


(d) How hard do you think your Desk Light resource baseline can be achieved?

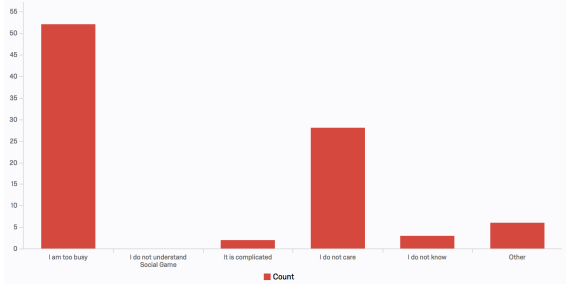
Figure 8.9: Spring semester survey questions regarding various resources baselines & relative difficulty to achieve—perform energy efficiently



(a) Since Social Game deployed, have you consciously performed energy-saving actions?



(b) **If yes:** What motivates your behaviors regarding your answer to the previous question?



(c) **If no:** What motivates your behaviors regarding your answer to the previous question?

Figure 8.10: Spring semester energy awareness question & what motivates occupants' energy efficient (or not) behavior

Chapter 9

Conclusion

This goal of this dissertation is the exploration of *Human-Centric Cyber-Physical Systems* by simultaneously considering user's behavior/preference and users interaction as strategic agents. We envision smart-building systems in which humans take control of shared or scarced resources, interact with sensing/actuating devices, and improve the environment they live in. Human's interaction in a cyber-physical system is a fundamental contribution and a core mechanism of the implementation of smart building technology. Generalized frameworks have been introduced for enabling humans to be involved in a closed loop cyber-physical system, learning their objective functions, and designing incentives to motivate their behavior. Our main goal is the development of applications in which humans are a part of smart infrastructure and interact in a close-loop control system with the cyber-physical system. Interestingly, such applications have great potential to improve sustainability and energy efficiency by building manager (or planner in general). In this context, the motivation for the work in this thesis was to address the following questions:

How can a building manager (or planner in general) enable humans in a closed-loop system? What are the motivations for human participation? What type of models/learning schemes emerge from the interaction between coupled decision-makers? What is the role of Artificial Intelligence in such systems?

This thesis demonstrates one of the first approaches toward modeling human interaction with cyber-physical systems in a multi-agent setting. Unsurprisingly, the true actions of agents and the manner in which these actions integrate with sensing/actuation platforms remain unknown to the planner tasked with improving operational efficiency. In Chapter 2, we derived the main framework for modeling the interaction between humans and other agents. In addition to the interaction model, a mechanism design step can be leveraged by the planner as closing—the—loop around decision making agents. However, a challenge is the learning task required for inferring such models. Machine learning algorithms and optimization theory techniques are used for the efficient modeling of agent interaction with cyber-physical systems. Our work in learning starts with a base learning scheme for non-cooperative continuous games (Chapter 3) and then extends to robust learning methods (Chapter 5),

probabilistic graphical adaptations (Chapter 6), and efficient learning approaches by leveraging agents' networking effects (Chapter 7). Vulnerabilities under cyber-attacks and data poisoning are discussed in depth in Chapter 4. In Chapter 8, Deep Learning methods are introduced for modeling temporal dependencies of agents' decision-making actions under a sequential discrete choice non-cooperative game.

9.1 Future Research Frontiers

As cyber-physical systems become critical infrastructures, more and more agents, buildings, and even cities (higher level entities) will be interacting and creating multi-level closed—loop control systems. For future work, we find it promising to follow several paths for cyber-physical systems combining several Machine Learning & Artificial Intelligence frameworks.

9.1.1 Human-Centric Cyber-Physical Systems & Smart Grid

Gamification and other game theoretic methods create a novel framework for combining agents, optimize on their preferences, which in most cases are unknown. Smart grids of the future will adapt numerous Machine Learning & Artificial Intelligence frameworks towards smart grid security and attack [75], fault detection at building equipment level (power converters) [137, 138] and broader smart-grid components like: wind turbines/farms [61, 97, 109, 110], network distribution systems [190, 191, 192], and in nanogrids [136]. Moreover, occupants' sensing in a smart building—grid [196, 199, 201] can lead/target towards sustainability. By integrating the developed *Human-Centric Cyber-Physical System* frameworks with several novel smart grid technologies, we introduce an interesting problem towards smart grid efficiency and building—grid integration/optimization. Recent advances in grid level [38, 82] along with energy retailer market and building level engagement [81] should be taken into account for an overall reliable structure. In this type of interconnected systems we can take advantage that game theory models complex interactions, which are valuable for several smart-grid components. Additional statistical learning methods for tasks like energy prediction [77] will open new horizons for smart grid and can also be utilized. Hence, motivated by these challenges in smart grids, *Human-Centric Cyber-Physical Systems* will reflect new interests and applications.

9.1.2 Human-Centric Cyber-Physical Systems & Smart Buildings

Involving humans and their interaction with cyber-physical systems creates complexity and uncertainty. The cooperation of human elements with building automation in smart infrastructure helps improve system robustness and sustainability through a combination of control and flexibility. Moreover, it may be capable to improve the service that it offers to building

occupants. This flexibility makes it possible to accommodate situations like automatic shifting or curtailing demand during peak hours. More broadly, the goal of many infrastructure systems is to enact system-level efficiency improvements by using a *high-level planner* (e.g. facility manager) to coordinate autonomously *acting agents* in the system (e.g. selfish human decision-makers). It is this type of functionality that makes smart building technology so essential to the development of an ideal *smart city*. *Human-Centric Cyber-Physical Systems* have a great potential either as design component of smart building novel architecture [67] or even as part of broader *societal-scale cyber-physical systems* [147]. To this extent, deeper learning frameworks can be developed [73, 80] with possible focus on safety [155]. Moreover, advanced incentive—mechanism design algorithms [145] are important in sustaining gamification applications and transforming them to a profitable application. Lastly, improvements in the incentive design can be developed using novel segmentation analysis [35]. In this way, agents' (e.g. building occupants) contributed features-factors characterizing their decision-making in competitive environments can efficiently be inferred towards more targeted incentive design approaches.

Bibliography

[1] M.H. Albadi and E.F. El-Saadany. "A summary of demand response in electricity markets". In: *Electric Power Systems Research* 78.11 (2008), pp. 1989–1996. ISSN: 0378-7796. DOI: 10.1016/j.epsr.2008.04.002.

[2] Scott Alfeld, Xiaojin Zhu, and Paul Barford. "Data Poisoning Attacks against Autoregressive Models". In: *AAAI*. 2016.

[3] Merlinda Andoni et al. "Game-theoretic modeling of curtailment rules and network investments with distributed generation". In: *Applied Energy* 201 (2017), pp. 174–187. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2017.05.035. URL: http://www.sciencedirect.com/science/article/pii/S030626191730541X.

[4] James B Ang. "CO2 emissions, energy consumption, and output in France". In: *Energy Policy* 35.10 (2007), pp. 4772–4778.

[5] Kenneth J Arrow, Leonid Hurwicz, and Hirofumi Uzawa. "Constraint qualifications in maximization problems". In: *Naval Research Logistics Quarterly* 8.2 (1961), pp. 175–191.

[6] A. Aswani et al. "Identifying models of HVAC systems using semi-parametric regression". In: *Proc. of the American Control Conf.* 2012, pp. 3675–3680.

[7] Robert J. Aumann. "Subjectivity and correlation in randomized strategies". In: *J. Mathematical Economics* 1.1 (1974), pp. 67–96.

[8] Magnus Bang, Carin Torstensson, and Cecilia Katzeff. "The PowerHouse: A Persuasive Computer Game Designed to Raise Awareness of Domestic Energy Consumption". In: ed. by Wijnand A. IJsselsteijn et al. Springer Berlin Heidelberg, 2006.

[9] Robert A Baron, Mark S Rea, and Susan G Daniels. "Effects of indoor lighting (illuminance and spectral distribution) on the performance of cognitive tasks and interpersonal behaviors: The potential mediating role of positive affect". In: *Motivation and emotion* 16.1 (1992), pp. 1–33.

[10] Gözen Başar and Chandra Bhat. "A parameterized consideration set model for airport choice: an application to the San Francisco Bay area". In: *Transportation Research Part B: Methodological* 38.10 (2004), pp. 889–904.

[11] Wim Bernasco and Richard Block. "Where offenders choose to attack: A discrete choice model of robberies in Chicago". In: *Criminology* 47.1 (2009), pp. 93–130.

[12] Steven Berry, James Levinsohn, and Ariel Pakes. "Automobile prices in market equilibrium". In: *Econometrica* 63.4 (1995), pp. 841–890.

[13] Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

[14] Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. "Data-driven estimation in equilibrium using inverse optimization". In: *Mathematical Programming* 153.2 (2015), pp. 595–633.

[15] Aaron Bestick et al. "An inverse correlated equilibrium framework for utility learning in multiplayer, noncooperative settings". In: *Proceedings of the 2nd ACM international conference on High confidence networked systems*. ACM. 2013, pp. 9–16.

[16] Battista Biggio, Blaine Nelson, and Pavel Laskov. "Poisoning attacks against support vector machines". In: *arXiv preprint arXiv:1206.6389* (2012).

[17] Battista Biggio et al. "Poisoning attacks to compromise face templates". In: *Biometrics (ICB), 2013 International Conference on*. IEEE. 2013, pp. 1–7.

[18] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[19] Patrick Bolton and Mathias Dewatripont. *Contract theory*. MIT press, 2005.

[20] Magnus Boman et al. "Energy saving and added customer value in intelligent buildings". In: *Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*. 1998, pp. 505–517.

[21] Denis Bourgeois, Christoph Reinhart, and Iain Macdonald. "Adding advanced behavioural models in whole building energy simulation: A study on the total energy impact of manual and automated lighting control". In: *Energy and Buildings* 38.7 (2006), pp. 814–823. DOI: 10.1016/j.enbuild.2006.03.002.

[22] Emmanuel J. Candes et al. "Robust principal component analysis". In: *Journal of the ACM* 58.3 (2011).

[23] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[24] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[25] Yudong Chen, Constantine Caramanis, and Shie Mannor. *Robust High Dimensional Sparse Regression and Matching Pursuit*. arXiv:1301.2725. 2013.

[26] Yudong Chen, Constantine Caramanis, and Shie Mannor. "Robust Sparse Regression under Adversarial Corruption". In: *Proc. International Conference on Machine Learning*. ICML. 2013.

[27] Ziyi Chen, Jinwen Ma, and Yatong Zhou. "A Precise Hard-Cut EM Algorithm for Mixtures of Gaussian Processes". In: *Intelligent Computing Methodologies: 10th International Conference, ICIC 2014, Taiyuan, China, August 3-6, 2014. Proceedings*. Ed. by De-Shuang Huang, Kang-Hyun Jo, and Ling Wang. Springer International Publishing, 2014, pp. 68–75. DOI: 10.1007/978-3-319-09339-0_7.

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

[29] Thomas F Coleman and Yuying Li. "A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables". In: *SIAM Journal on Optimization* 6.4 (1996), pp. 1040–1058.

[30] Thomas H Cormen et al. *Introduction to algorithms*. MIT press, 2009.

[31] Ben Cowley et al. "Learning principles and interaction design for 'Green My Place': A massively multiplayer serious game". In: *Entertainment Computing* 2.2 (2011), pp. 103–113.

[32] Gabriela F. Cretu-Ciocarlie et al. "Casting out Demons: Sanitizing Training Data for Anomaly Sensors". In: *Proc. IEEE Security and Privacy Symposium*. S&P. 2008.

[33] Francisco Cribari-Neto. "Asymptotic inference under heteroskedasticity of unknown form". In: *Computational Statistics & Data Analysis* 45.2 (2004), pp. 215–233.

[34] Matthew Crouse and Richard G Baraniuk. "Contextual hidden Markov models for wavelet-domain signal processing". In: *Asilomar Conference on Signals, Systems, and Computers*. Vol. 1. 1997, pp. 95–100.

[35] Hari Prasanna Das et al. "Segmentation Analysis in Human Centric Cyber-Physical Systems using Graphical Lasso". In: *arXiv preprint arXiv:1810.10533* (2018).

[36] Vanessa De Luca and Roberta Castri. "The social power game: A smart application for sharing energy-saving behaviours in the city". In: *FSEA 2014* 27 (2014).

[37] Sebastian Deterding et al. "From game design elements to gamefulness: defining gamification". In: *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. ACM. 2011, pp. 9–15.

[38] Roel Ignatius Jacobus Dobbe. "An Integrative Approach to Data-Driven Monitoring and Control of Electric Distribution Networks". PhD thesis. UC Berkeley, 2018.

[39] Ivan Evtimov et al. "Robust Physical-World Attacks on Machine Learning Models". In: *arXiv preprint arXiv:1707.08945* (2017).

[40] Cheng Fan, Fu Xiao, and Yang Zhao. "A short-term building cooling load prediction method using deep learning algorithms". In: *Applied energy* 195 (2017), pp. 222–233.

[41] Guoliang Fan and Xiang-Gen Xia. "Image denoising using a local contextual hidden Markov model in the wavelet domain". In: *IEEE signal processing letters* 8.5 (2001), pp. 125–128.

[42] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern recognition letters* 27.8 (2006), pp. 861–874.

[43] Sjur D Flåm. "Solving non-cooperative games by continuous subgradient projection methods". In: *System Modelling and Optimization*. Springer, 1990, pp. 115–123.

[44] David A. Freedman. "Statistical models: theory and practice". In: Cambridge University Press, 2009.

[45] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Ann. Statistics* 29.5 (2001), pp. 1189–1232.

[46] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, NY, USA: 2001.

[47] Monika Frontczak et al. "Quantitative relationships between occupant satisfaction and satisfaction aspects of indoor environmental quality and building design". In: *Indoor air* 22.2 (2012), pp. 119–131.

[48] Drew Fudenberg et al. *The theory of learning in games*. Vol. 2. MIT press, 1998.

[49] S.A. Gabriel et al. *Complementarity Modeling in Energy Markets*. Int. Series Operations Research & Management Science. Springer-Verlag New York, 2013.

[50] Michel Gendreau. "On the location of eigenvalues of off-diagonal constant matrices". In: *Linear Algebra and its Applications* 79 (1986), pp. 99–102. DOI: 10.1016/0024-3795(86)90294-6.

[51] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[52] Philip E Gill, Walter Murray, and Margaret H Wright. "Practical optimization". In: (1981).

[53] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. arXiv:1412.6572. 2014.

[54] Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.

[55] Azucena Gracia and Tiziana de Magistris. "The demand for organic foods in the South of Italy: A discrete choice model". In: *Food Policy* 33.5 (2008), pp. 386–396.

[56] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks". In: *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE. 2013, pp. 6645–6649.

[57] Karol Gregor et al. "Draw: A recurrent neural network for image generation". In: *arXiv preprint arXiv:1502.04623* (2015).

[58] N. Groot, B. De Schutter, and H. Hellendoorn. "Reverse Stackelberg games, Part I: Basic framework". In: *IEEE International Conference on Control Applications,* Oct. 2012, pp. 421–426. DOI: 10.1109/CCA.2012.6402334.

[59] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[60] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[61] R. Lily Hu et al. "Using Domain Knowledge Features for Wind Turbine Diagnostics". In: *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2016.

[62] Peter J Huber. *Robust statistics*. Springer, 2011.

[63] Clifford M Hurvich, Jeffrey S Simonoff, and Chih-Ling Tsai. "Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion". In: *J. Royal Statistical Society: Series B (Statistical Methodology)* 60.2 (1998), pp. 271–293.

[64] Rob J Hyndman and Anne B Koehler. "Another look at measures of forecast accuracy". In: *Inter. J. forecasting* 22.4 (2006), pp. 679–688.

[65] Rob J Hyndman and Anne B Koehler. "Another look at measures of forecast accuracy". In: *International journal of forecasting* 22.4 (2006), pp. 679–688.

[66] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[67] Ruoxi Jia et al. "Design Automation for Smart Building Systems". In: *Proceedings of the IEEE* 106.9 (2018), pp. 1680–1699.

[68] Ruoxi Jia et al. "MapSentinel: Can the Knowledge of Space Use Improve Indoor Tracking Further?" In: *Sensors* 16.4 (2016), p. 472.

[69] Ruoxi Jia et al. "Poisoning Attacks on Data-Driven Utility Learning in Games". In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018, pp. 5774–5780.

[70] Ruoxi Jia et al. "Privacy-Enhanced Architecture for Occupancy-based HVAC Control". In: *arXiv preprint arXiv:1607.03140* (2016).

[71] Ruoxi Jia et al. "SoundLoc: Accurate Room-level Indoor Localization using Acoustic Signatures". In: *IEEE International Conference on Automation Science and Engineering (IEEE CASE 2015)*. 2015, pp. 186–193.

[72] Xiaofan Jiang et al. "Design and implementation of a high-fidelity ac metering network". In: *Proc. Inter. Conf. on Information Processing in Sensor Networks*. IEEE. 2009, pp. 253–264.

[73] Ming Jin. "Data-efficient Analytics for Optimal Human-Cyber-Physical Systems". PhD thesis. UC Berkeley, 2017.

[74] Ming Jin, Ruoxi Jia, and Costas J Spanos. "Virtual occupancy sensing: Using smart meters to indicate your presence". In: *IEEE Transactions on Mobile Computing* 16.11 (2017), pp. 3264–3277.

[75] Ming Jin, Javad Lavaei, and Karl Johansson. "A Semidefinite Programming Relaxation under False Data Injection Attacks against Power Grid AC State Estimation". In: *55th Annual Allerton Conference on Communication, Control, and Computing*. 2017.

[76] Ming Jin, Javad Lavaei, and Karl H Johansson. "Power Grid AC-based State Estimation: Vulnerability Analysis Against Cyber Attacks". In: *IEEE Transactions on Automatic Control* (2018).

[77] Ming Jin and Costas Spanos. "BRIEF: Bayesian Regression of Infinite Expert Forecasters for Single and Multiple Time Series Prediction". In: *54th IEEE Conference on Decision and Control (CDC 2015)*. 2015, pp. 78–83.

[78] Ming Jin et al. "Automated mobile sensing: Towards high-granularity agile indoor environmental quality monitoring". In: *Building and Environment* 127 (2018), pp. 268–276.

[79] Ming Jin et al. "Environmental Sensing by Wearable Device for Indoor Activity and Location Estimation". In: *40th Annual Conference of the IEEE Industrial Electronics Society (IECON 2014)*. 2014, pp. 5369–5375.

[80] Ming Jin et al. "Inverse Reinforcement Learning via Deep Gaussian Process". In: *The Conference on Uncertainty in Artificial Intelligence (UAI)*. 2017.

[81] Ming Jin et al. "Microgrid to enable optimal distributed energy retail and end-user demand response". In: *Applied Energy* 210 (2018), pp. 1321–1335. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2017.05.103. URL: http://www.sciencedirect.com/science/article/pii/S0306261917306062.

[82] Ming Jin et al. "MOD-DR: Microgrid optimal dispatch with demand response". In: *Applied Energy* 187 (2017), pp. 758–776. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2016.11.093. URL: http://www.sciencedirect.com/science/article/pii/S030626191631724X.

[83] Ming Jin et al. "Occupancy detection via environmental sensing". In: *IEEE Transactions on Automation Science and Engineering* 15.2 (2018), pp. 443–455.

[84] Ming Jin et al. "PresenceSense: Zero-training Algorithm for Individual Presence Detection based on Power Monitoring". In: *Proc. 1st ACM Conf. Embedded Systems for Energy–Efficient Buildings*. 2014, pp. 1–10.

[85] Ming Jin et al. "REST: a reliable estimation of stopping time algorithm for social game experiments". In: *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. ACM. 2015, pp. 90–99.

[86] Ming Jin et al. "Sensing by proxy: Occupancy detection based on indoor CO2 concentration". In: *The 9th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'15)*. 2015, pp. 1–14.

[87] Michael I Jordan and Robert A Jacobs. "Hierarchical mixtures of experts and the EM algorithm". In: *Neural computation* 6.2 (1994), pp. 181–214.

[88] Caroline Karmann, Stefano Schiavon, and Edward Arens. "Percentage of commercial buildings showing at least 80% occupant satisfied with their thermal comfort". In: (2018).

[89] Michael Kearns and Ming Li. "Learning in the presence of malicious errors". In: *SIAM Journal on Computing* 22.4 (1993), pp. 807–837.

[90] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *Proceedings of the IEEE International Conference on Neural Networks*. Vol. 4. Nov. 1995, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968.

[91] James Kennedy. "Particle swarm optimization". In: *Encyclopedia of Machine Learning*. Springer, 2010, pp. 760–766.

[92] Arezou Keshavarz, Yang Wang, and Stephen Boyd. "Imputing a convex objective function". In: *IEEE Intern. Symp. on Intelligent Control*. 2011, pp. 613–619.

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

[94] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[95] Marius Kloft and Pavel Laskov. "Security analysis of online centroid anomaly detection". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 3681–3724.

[96] Erik Knol and Peter W de Vries. "EnerCities—A Serious Game to Stimulate Sustainability and Energy Conservation: Preliminary Results". In: *eLearning Papers* 25 (2011).

[97] Ioannis C Konstantakopoulos, Michael K Bourdoulis, and Antonio T Alexandridis. "An Alternative PI Controller Design Approach for PWM-regulated ac/dc three-phase Converters". In: *Industrial Technology (ICIT), 2012 IEEE International Conference on*. IEEE. 2012, pp. 944–949.

[98] Ioannis C Konstantakopoulos et al. "A Deep Learning and Gamification Approach to Energy Conservation at Nanyang Technological University". In: *arXiv preprint arXiv:1809.05142* (2018).

[99] Ioannis C Konstantakopoulos et al. "A robust utility learning framework via inverse optimization". In: *IEEE Transactions on Control Systems Technology* 26.3 (2018), pp. 954–970.

[100] Ioannis C Konstantakopoulos et al. "Inverse modeling of non-cooperative agents via mixture of utilities". In: *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE. 2016, pp. 6327–6334.

[101] Ioannis C Konstantakopoulos et al. "Leveraging correlations in utility learning". In: *American Control Conference (ACC), 2017*. IEEE. 2017, pp. 5249–5256.

[102] Ioannis C. Konstantakopoulos et al. "Smart Building Energy Efficiency via Social Game: A Robust Utility Learning Framework for Closing–the–Loop". In: *Proc. 1st Intern. Workshop on Science of Smart City Operations and Platforms Engineering*. 2016.

[103] Ioannis C Konstantakopoulos et al. "Social game for building energy efficiency: Utility learning, simulation, and analysis". In: *arXiv preprint arXiv:1407.0727* (2014).

[104] Ioannis Konstantakopoulos, Costas J Spanos, and S Shankar Sastry. *Social game for building energy efficiency: Utility learning, simulation, analysis and incentive design*. Tech. rep. Technical Report UCB/EECS-2015-3, EECS Department, University of California, Berkeley, 2015.

[105] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[106] Jean-Jacques Laffont and David Martimort. *The Theory of Incentives: The Principal–Agent Model*. Princeton University Press, 2002.

[107] John A. Laitner. "Energy efficiency: rebounding to a sound analytical perspective". In: *Energy Policy* 28.6–7 (2000), pp. 471–475. DOI: 10.1016/S0301-4215(00)00032-X.

[108] Wing-Kai Lam, Fai Yung, and Lei Xu. "An experimental comparative study on several soft and hard-cut EM algorithms for mixture of experts". In: *International Conference on Neural Networks*. Vol. 3. 1997, pp. 1574–1579. DOI: 10.1109/ICNN.1997.614128.

[109] Kevin Leahy et al. "Diagnosing and predicting wind turbine faults from SCADA data using support vector machines". In: *Prognostics and Health Management Society* (2018).

[110] Kevin Leahy et al. "Diagnosing wind turbine faults using machine learning techniques applied to operational data". In: *Prognostics and Health Management (ICPHM), 2016 IEEE International Conference on. IEEE*. 2016.

[111] Honglak Lee et al. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations". In: *Proceedings of the 26th annual international conference on machine learning*. ACM. 2009, pp. 609–616.

[112] Bo Li and Yevgeniy Vorobeychik. "Feature cross-substitution in adversarial classification". In: *Advances in Neural Information Processing Systems*. NIPS. 2014, pp. 2087–2095.

[113] Bo Li and Yevgeniy Vorobeychik. "Scalable Optimization of Randomized Operational Decisions in Adversarial Classification Settings". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. 2015, pp. 599–607.

[114] Bo Li et al. "Data poisoning attacks on factorization-based collaborative filtering". In: *Advances in neural information processing systems*. 2016, pp. 1885–1893.

[115] 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*. Aug. 2014, pp. 407–412. DOI: 10.1109/CoASE.2014.6899357.

[116] Brantley Liddle and Sidney Lung. "Revisiting energy consumption and GDP causality: Importance of a priori hypothesis testing, disaggregated data, and heterogeneous panels". In: *Applied Energy* 142 (2015), pp. 44–55.

[117] Yanpei Liu et al. "Delving into transferable adversarial examples and black-box attacks". In: *arXiv preprint arXiv:1611.02770* (2016).

[118] Y. Ma, G. Anderson, and F. Borrelli. "A distributed predictive control approach to building temperature regulation". In: *Proc. of the American Control Conf.* 2011, pp. 2089–2094.

[119] Jeffrey K MacKie-Mason and Hal R Varian. "Generalized vickrey auctions". In: (1994).

[120] Sabita Maharjan et al. "Dependable demand response management in the smart grid: A Stackelberg game approach". In: *IEEE Transactions on Smart Grid* 4.1 (2013), pp. 120–132.

[121] Milos Manic et al. "Intelligent buildings of the future: Cyberaware, deep learning powered, and human interacting". In: *IEEE Industrial Electronics Magazine* 10.4 (2016), pp. 32–49.

[122] J.L. Mathieu et al. "Using Residential Electric Loads for Fast Demand Response: The Potential Resource and Revenues, the Costs, and Policy Recommendations". In: *ACEEE Summer Study on Energy Efficiency in Buildings*. 2012.

[123] Johanna L Mathieu et al. "Quantifying changes in building electricity use, with application to demand response". In: *IEEE Transactions on Smart Grid,* 2.3 (2011), pp. 507–518.

[124] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[125] JM McQuade. "A system approach to high performance buildings". In: *United Technologies Corporation, Tech. Rep* (2009).

[126] Shike Mei and Xiaojin Zhu. "The Security of Latent Dirichlet Allocation". In: *AISTATS*. 2015.

[127] Deepak Merugu, Balaji S Prabhakar, and NS Rama. "An incentive mechanism for decongesting the roads: A pilot program in Bangalore". In: *Proceedings of ACM NetEcon Workshop*. 2009.

[128] Elena Mocanu et al. "Deep learning for estimating building energy consumption". In: *Sustainable Energy, Grids and Networks* 6 (2016), pp. 91–99.

[129] John Nash. "Non-cooperative games". In: *Annals of mathematics* (1951), pp. 286–295.

[130] John F Nash et al. "Equilibrium points in n-person games". In: *Proceedings of the national academy of sciences* 36.1 (1950), pp. 48–49.

[131] Noam Nisan et al. *Algorithmic game theory*. Cambridge University Press, 2007.

[132] Brian Orland et al. "Saving energy in an office environment: A serious game intervention". In: *Energy and Buildings* 74 (2014), pp. 43–52. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2014.01.036. URL: http://www.sciencedirect.com/science/article/pii/S0378778814000747.

[133] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[134] Hanchuan Peng, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on pattern analysis and machine intelligence* 27.8 (2005), pp. 1226–1238.

[135] Christopher Pluntke and Balaji Prabhakar. "INSINC: A Platform for Managing Peak Demand in Public Transit". In: *JOURNEYS, Land Transport Authority Academy of Singapore* (2013), pp. 31–39.

[136] Jason Poon et al. "FailSafe: A generalized methodology for converter fault detection, identification, and remediation in nanogrids". In: *Building Efficiency and Sustainable Technologies, 2015 IEEE International Conference on*. IEEE. 2015, pp. 73–78.

[137] Jason Poon et al. "Model-based fault detection and identification for switching power converters". In: *IEEE Transactions on Power Electronics* 32.2 (2017), pp. 1419–1430.

[138] Jason Poon et al. "Real-time model-based fault diagnosis for switching power converters". In: *Applied Power Electronics Conference and Exposition (APEC), 2015 IEEE*. IEEE. 2015, pp. 358–364.

[139] Ariadna Quattoni et al. "Hidden conditional random fields". In: *IEEE transactions on pattern analysis and machine intelligence* 29.10 (2007).

[140] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv:1511.06434* (2015).

[141] Sarvapali D Ramchurn et al. "Agent-based control for decentralised demand side management in the smart grid". In: *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems. 2011, pp. 5–12.

[142] Lillian J Ratliff, Samuel A Burden, and S Shankar Sastry. "Characterization and computation of local nash equilibria in continuous games". In: *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*. IEEE. 2013, pp. 917–924.

[143] Lillian J. Ratliff, Samuel A. Burden, and S. Shankar Sastry. "Generictiy and Structural Stability of Non–Degenerate Differential Nash Equilibria". In: *Proc. 2014 American Controls Conf.* 2014.

[144] Lillian J. Ratliff, Samuel A. Burden, and S. Shankar Sastry. "On the Characterization of Local Nash Equilibria in Continuous Games". In: *IEEE Transactions on Automatic Control* (2016 (to appear).).

[145] Lillian J Ratliff and Tanner Fiez. "Adaptive Incentive Design". In: *arXiv preprint arXiv:1806.05749* (2018).

[146] Lillian J Ratliff and Eric Mazumdar. "Risk-Sensitive Inverse Reinforcement Learning via Gradient Methods". In: *arXiv preprint arXiv:1703.09842* (2017).

[147] Lillian Jane Ratliff. "Incentivizing efficiency in societal-scale cyber-physical systems". PhD thesis. UC Berkeley, 2015.

[148] Lillian J Ratliff et al. "Incentive Design and Utility Learning via Energy Disaggregation". In: *Proc. of the 19th World Congress of the Inter. Federation of Automatic Control*. 2014.

[149] Lillian J Ratliff et al. "Social game for building energy efficiency: Incentive design". In: *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*. IEEE. 2014, pp. 1011–1018.

[150] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *arXiv preprint arXiv:1401.4082* (2014).

[151] M. Roozbehani, M. Dahleh, and S. Mitter. "Dynamic Pricing and Stabilization of Supply and Demand in Modern Electric Power Grids". In: *First IEEE International Conference on Smart Grid Communications*. Oct. 2010, pp. 543–548.

[152] J Ben Rosen. "Existence and uniqueness of equilibrium points for concave n-person games". In: *Econometrica: Journal of the Econometric Society* (1965), pp. 520–534.

[153] Benjamin IP Rubinstein et al. "Antidote: understanding and defending against poisoning of anomaly detectors". In: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM. 2009, pp. 1–14.

[154] Erica E Ryherd and Lily M Wang. "Implications of human performance and perception under tonal noise conditions on indoor noise criteria". In: *The Journal of the Acoustical Society of America* 124.1 (2008), pp. 218–226.

[155] Dorsa Sadigh. "Safe and Interactive Autonomy: Control, Learning, and Verification". PhD thesis. UC Berkeley, 2017.

[156] Vahid Salehi et al. "Laboratory-based smart power system, part I: Design and system development". In: *IEEE Transactions on Smart Grid* 3.3 (2012), pp. 1394–1404.

[157] Ricardo Salvador, Teresa Romão, and Pedro Centieiro. "A Gesture Interface Game for Energy Consumption Awareness". In: *Proc. 9th Int. Conf. Advances in Computer Entertainment*. 2012, pp. 352–367. DOI: 10.1007/978-3-642-34292-9_25.

[158] Stan Salvador and Philip Chan. "Toward accurate dynamic time warping in linear time and space". In: *Intelligent Data Analysis* 11.5 (2007), pp. 561–580.

[159] Pedram Samadi et al. "Advanced demand side management for the future smart grid using mechanism design". In: *IEEE Transactions on Smart Grid* 3.3 (2012), pp. 1170–1180.

[160] Lee Schipper and Michael Grubb. "On the rebound? Feedback between energy intensities and energy uses in IEA countries". In: *Energy Policy* 28.6–7 (2000), pp. 367–388. DOI: 10.1016/S0301-4215(00)00018-5.

[161] Claude E Shannon. "Communication theory of secrecy systems". In: *Bell Labs Technical Journal* 28.4 (1949), pp. 656–715.

[162] Jonathan Simon, Marco Jahn, and Amro Al-Akkad. "Saving Energy at Work: The Design of a Pervasive Game for Office Spaces". In: *Proc. 11th Int. Conf. Mobile and Ubiquitous Multimedia*. 2012. DOI: 10.1145/2406367.2406379.

[163] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[164] Cristian Sminchisescu, Atul Kanaujia, and Dimitris Metaxas. "Conditional models for contextual human motion recognition". In: *Computer Vision and Image Understanding* 104.2-3 (2006), pp. 210–220.

[165] Adriaan R Soetevent and Peter Kooreman. "A discrete-choice model with social interactions: with an application to high school teen behavior". In: *Journal of Applied Econometrics* 22.3 (2007), pp. 599–624.

[166] Z Song. "Collaborative Building Control to Optimize Energy Saving and Improve Occupants' Experience". In: *ASHRAE Trans.* 119.AA1 (2013).

[167] Nitish Srivastava et al. "Dropout: A simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[168] Jan Sundell et al. "Ventilation rates and health: multidisciplinary review of the scientific literature". In: *Indoor air* 21.3 (2011), pp. 191–204.

[169] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. "Deep neural networks for object detection". In: *Advances in neural information processing systems*. 2013, pp. 2553–2561.

[170] Genichi Taguchi, Elsayed A Elsayed, and Thomas C Hsiang. *Quality engineering in production systems*. McGraw-Hill College, 1989.

[171] Robert Tibshirani and Keith Knight. "Model search and inference by bootstrap "bumping"". In: *J. Comp. and Graph. Statistics* 8.4 (1999), pp. 671–686.

[172] Kenneth Train. "A validation test of a disaggregate mode choice model". In: *Transportation Research* 12.3 (1978), pp. 167–174.

[173] Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

[174] Amos Tversky and Daniel Kahneman. "The framing of decisions and the psychology of choice". In: *Environmental Impact assessment, technology assessment, and risk analysis*. Springer, 1985, pp. 107–129.

[175] David E Tyler. "Robust statistics: Theory and methods". In: *Journal of the American Statistical Association* 103.482 (2008), pp. 888–889.

[176] Gerhard Venter and Jaroslaw Sobieszczanski-Sobieski. "Particle swarm optimization". In: *AIAA journal* 41.8 (2003), pp. 1583–1589.

[177] Oriol Vinyals et al. "Show and tell: A neural image caption generator". In: *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE. 2015, pp. 3156–3164.

[178] José Vuelvas, Fredy Ruiz, and Giambattista Gruosso. "Limiting gaming opportunities on incentive-based demand response programs". In: *Applied Energy* 225 (2018), pp. 668–681. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.05.050. URL: http://www.sciencedirect.com/science/article/pii/S0306261918307608.

[179] Pawel Wargocki et al. "Indoor climate and productivity in offices. How to integrate productivity in life cycle costs analysis of building services". In: (2008).

[180] Greg CG Wei and Martin A Tanner. "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms". In: *Journal of the American statistical Association* 85.411 (1990), pp. 699–704.

[181] Max Welling and Yee W Teh. "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 681–688.

[182] Yonghui Wu et al. "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144* (2016).

[183] Huang Xiao et al. "Is feature selection secure against training data poisoning?" In: *International Conference on Machine Learning*. 2015, pp. 1689–1698.

[184] Huan Xu, Constantine Caramanis, and Shie Mannor. "Robust Regression and Lasso". In: *IEEE Transactions on Information Theory* 56.7 (2010), pp. 3561–3574.

[185] Kelvin Xu et al. "Show, attend and tell: Neural image caption generation with visual attention". In: *International Conference on Machine Learning*. 2015, pp. 2048–2057.

[186] Yong Xu et al. "A regression approach to speech enhancement based on deep neural networks". In: *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 23.1 (2015), pp. 7–19.

[187] Mengmeng Yu and Seung Ho Hong. "Incentive-based demand response considering hierarchical electricity market: A Stackelberg game approach". In: *Applied Energy* 203 (2017), pp. 267–279. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2017.06.010. URL: http://www.sciencedirect.com/science/article/pii/S0306261917307602.

[188] Mengmeng Yu and Seung Ho Hong. "Supply-demand balancing for power management in smart grid: A Stackelberg game approach". In: *Applied Energy* 164 (2016), pp. 702–710. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2015.12.039. URL: http://www.sciencedirect.com/science/article/pii/S0306261915016128.

[189] Ni Zhang, Yu Yan, and Wencong Su. "A game-theoretic economic operation of residential distribution system with high participation of distributed electricity prosumers". In: *Applied Energy* 154 (2015), pp. 471–479. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2015.05.011. URL: http://www.sciencedirect.com/science/article/pii/S0306261915006212.

[190] Yuxun Zhou et al. "Abnormal event detection with high resolution micro-PMU data". In: *Power Systems Computation Conference (PSCC), 2016*. IEEE. 2016, pp. 1–7.

[191] Yuxun Zhou et al. "Data-driven event detection with partial knowledge: A hidden structure semi-supervised learning method". In: *American Control Conference (ACC), 2016*. IEEE. 2016, pp. 5962–5968.

[192] Yuxun Zhou et al. "Distribution Network Event Detection with Ensembles of Bundle Classifiers". In: *IEEE PES General Meeting 2016*. 2016.

[193] Lijing Zhu et al. "Study on crowdfunding's promoting effect on the expansion of electric vehicle charging piles based on game theory analysis". In: *Applied Energy* 196 (2017), pp. 238–248. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2016.11.060. URL: http://www.sciencedirect.com/science/article/pii/S0306261916316452.

[194] Han Zou et al. "Adaptive Localization in Dynamic Indoor Environments by Transfer Kernel Learning". In: *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*. IEEE. 2017, pp. 1–6.

[195] Han Zou et al. "An integrative weighted path loss and extreme learning machine approach to rfid based indoor positioning". In: *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE. 2013, pp. 1–5.

[196] Han Zou et al. "Device-free occupancy detection and crowd counting in smart buildings with WiFi-enabled IoT". In: *Energy and Buildings* 174 (2018), pp. 309–322. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2018.06.040. URL: http://www.sciencedirect.com/science/article/pii/S0378778817339336.

[197] Han Zou et al. "Non-intrusive occupancy sensing in commercial buildings". In: *Energy and Buildings* 154 (2017), pp. 633–643. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2017.08.045. URL: http://www.sciencedirect.com/science/article/pii/S0378778816311987.

[198] Han Zou et al. "Standardizing location fingerprints across heterogeneous mobile devices for indoor localization". In: *2016 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2016, pp. 1–6.

[199] Han Zou et al. "Towards occupant activity driven smart buildings via WiFi-enabled IoT devices and deep learning". In: *Energy and Buildings* 177 (2018), pp. 12–22. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2018.08.010. URL: http://www.sciencedirect.com/science/article/pii/S037877881831329X.

[200] Han Zou et al. "Unsupervised WiFi-enabled IoT Device-User Association for Personalized Location-based Service". In: *IEEE Internet of Things Journal* (2018).

[201] Han Zou et al. "WiFi-Based Human Identification via Convex Tensor Shapelet Learning." In: *AAAI*. 2018.

[202] Han Zou et al. "WinIPS: WiFi-Based Non-Intrusive Indoor Positioning System With Online Radio Map Construction and Adaptation". In: *IEEE Transactions on Wireless Communications* 16.12 (2017), pp. 8118–8130.

[203] Han Zou et al. "WinLight: A WiFi-based occupancy-driven lighting control system for smart building". In: *Energy and Buildings* 158 (2018), pp. 924–938. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2017.09.001. URL: http://www.sciencedirect.com/science/article/pii/S0378778817313907.