The Dissertation Committee for Yinshuang Xiao
certifies that this is the approved version of the following dissertation:

Socio-Technical Systems Engineering and Design: A Meso-Level Network-Based Approach

Committee:

Zhenghui Sha, Supervisor

Carolyn Seepersad

Richard Crawford

Ming Zhang

Socio-Technical Systems Engineering and Design: A Meso-Level Network-Based Approach

by

Yinshuang Xiao

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2024

Acknowledgments

First and foremost, I am profoundly grateful to my advisor, Prof. Zhenghui Sha, for his unwavering support, guidance, and mentorship throughout my doctoral journey. His expertise, encouragement, and insightful feedback have been invaluable in shaping my research and personal growth. I extend my sincere appreciation to the members of my dissertation committee, Prof. Carolyn Seepersad, Prof. Richard Crawford, and Prof. Ming Zhang, for their constructive criticism, valuable suggestions, and commitment to academic excellence.

I express my sincere gratitude to my project advisors, Prof. Wei Chen, Prof. Noshir Contractor, Prof. Faez Ahmed, Prof. Johan Koskinen, Dr. Hyongju Park, and Dr. Yan Fu, whose insightful suggestions greatly enriched my work. My sincere appreciation also extends to collaborators like Dr. Yaxin Cui, Neelam Jignesh Modi, and Jonathan Haris Januar, whose collaboration and discussions enriched my doctoral experience.

I wish to extend my gratitude to my peers at the SiDi Lab, including Xingang Li, Siyu Chen, Yuewan Sun, Pawornwan (Bam) Thongmak, Phillip Gavino, Cole Mensch, John Clay, Michael Cardone, Samantha Montemayor, Anuj Swaminathan, Elisa Koolman, Laxmi Poudel, Molla Hafizur Rahman, Daniel Weber, Sumaiya Sultana Tanu, and Jared Poe. The collaborative and intellectually stimulating environment we fostered, characterized by mutual respect and a shared passion for research, has played a pivotal role in shaping my academic journey. I am also profoundly grateful for the unwavering support of my friends, particularly Xiaotong Sun, Qingyu Xiao, Yuchen Li, and all those who have continuously offered me moral encouragement and support.

My heartfelt thanks go to my family for their unwavering love, encouragement, and understanding throughout this journey. Their unwavering support and sacrifices have been the cornerstone of my academic endeavors.

I would like to acknowledge the funding agencies, namely the National Science Foundation (NSF), Division of Civil, Mechanical, and Manufacturing Innovation Grants #2005661 and #2203080, and the consortium of Cooperative Mobility for Competitive Megaregions associated with the Department of Transportation (DOT), whose financial support made this research possible. Additionally, I extend my special acknowledgment to the University of Arkansas and the University of Texas at Austin for their unwavering support, including financial support, mentorship from experienced faculty members, and valuable professional development opportunities, which have all played a crucial role in shaping this dissertation.

Lastly, I extend my deepest appreciation to all those who directly or indirectly contributed to this dissertation, including friends, mentors, and well-wishers. Your encouragement, belief in my abilities, and shared enthusiasm for academic inquiry have been a constant source of inspiration.

Abstract

Socio-Technical Systems Engineering and Design: A Meso-Level Network-Based Approach

Yinshuang Xiao, PhD
The University of Texas at Austin, 2024

SUPERVISOR: Zhenghui Sha

Different from traditional engineering systems design, which is keen on the design and optimization of technical artifacts, the design of socio-technical systems (STS) is guided by fundamentally understanding the complex interactions between social and technical aspects. This has posed significant challenges when applying existing systems engineering (SE) and design approaches to STS. For example, classical top-down design methodologies, such as the Waterfall model and the SE Vee model, are not appropriate for the engineering and design of large-scale STS with spontaneous interactions among individual entities or components. Although existing bottom-up design approaches are adaptable to the system scale, they primarily focus on understanding the behaviors and interactions between individual entities at the micro-level and their impact on system performance at the macro-level. In my dissertation research, the central hypothesis is that subsystems at the meso level (*e.g.*, small clusters of individual entities) serve as critical links in system structures and could influence both macro-level performance and micro-level interactions, and thus deserve scientific investigation in STS engineering and design. However, there is a knowledge gap in understanding what meaningful subsystem information at the meso-level is and how it can be extracted and used to guide the design of an STS system to achieve the desired system performance.

To fill this gap, my research objective is to develop a novel meso-level network-based framework for STS engineering and design. This dissertation is driven by answering three *research questions*: 1) *RQ1*: How can significant meso-level system structures be identified? 2) *RQ2*: What are the influences of the significant meso-level subsystems on the system performance at the macro level and the interaction mechanism at the micro level? 3) *RQ3*: How can meso-level structural information be used to design an STS to achieve desired macro-level performance and micro-level functionality? The methodologies proposed to address these questions are validated through two case studies: shared mobility systems and customer-product market systems. For shared mobility systems, a network motif-based robust design framework is proposed to improve the robustness and resilience of socio-technical systems against seasonal effects. Within this framework, trip motif mining addresses *RQ1*, while trip motif-based system robustness metrics tackle *RQ2*. Formulating and solving optimization problems serves to address *RQ3*. Additionally, a graph neural network-based (GNN-based) link prediction (LP) model is introduced to support STS design decision-making and validation. The GNN-based model leverages local network information to enhance prediction accuracy, addressing *RQ2*, while implementing the LP model for design strategy validation contributes to addressing *RQ3*.

In the context of customer-product market systems, a socio-technical system data collection framework integrating information retrieval and survey design methods is proposed to tackle the data scarcity issue in STSs. Furthermore, a novel micro-level entity design framework of STS, considering meso-level dependencies, is proposed, marking the first attempt to solve the inverse problem. This framework contributes to addressing *RQ1*, *RQ2*, and *RQ3* by incorporating network motif mining, quantification of subsystem-based individual entity functionality, and entity optimization design within a unified framework. Lastly, a preliminary exploration of meso-level temporal network motifs in STS is conducted, encompassing solutions to the dynamic data scarcity issue, dynamic network modeling, and significant temporal subnetwork mining and empirical interpretation. This exploration contributes to

answering *RQ1* and *RQ2* when considering the time dimension.

Regarding the key findings and conclusions of this dissertation, we first show the effectiveness of combining information retrieval and survey design to tackle the data accessibility challenge in STS network data. Additionally, our survey study, for the first time, gathered customers' social network data alongside their purchase decision-making data, aiding in the examination of social factors influencing customers' decision-making. Moreover, while survey studies are time-consuming, leveraging named entity recognition (NER) models for mining online text data offers a viable alternative for supporting entity relationship data collection. Then, when working on the shared mobility system case study, we find that: 1) An STS's seasonal sensitivity is closely tied to imbalanced capacity planning within its subsystems. Therefore, balancing the capacity of meso-level service systems is beneficial to enhancing STS robustness against seasonal demand fluctuations; 2) The outperformance of the GNN-based predictive model, which incorporates local network information, compared to a simple neural network model lacking such consideration, demonstrates the importance of local network information in demand prediction between stations in shared mobility networks. Moreover, this outperformance persists even when network structures and density change significantly. Next, in the study of design for customer-product systems, the inter-brand triadic competition closure competition, where three products from different brands form a closed triangle competition, emerges as a significant pattern in the vacuum cleaner market system. Identifying these meso-level patterns offers a means to quantify product competitiveness. Integrating this information with network predictive models and metaheuristic approaches, like the genetic algorithm, facilitates the inclusion of local competition data in the product design process. In the study of STS dynamic analysis, we demonstrate that increasing undersampling ratios improves predictive performance, particularly in moderately imbalanced systems, enhancing the GNN-based LP model. However, in extremely imbalanced systems, a tuning process is necessary to balance computational efficiency and model performance, with the threshold-based postprocessing method consistently outper-

forming the rank-based method. Additionally, six temporal competition motifs are interpreted, aiding in tracking market system dynamics.

In summary, my dissertation contributes to the systems science literature by introducing a novel meso-level network-based framework for STS engineering and design, thus addressing the knowledge gap pertaining to the identification and interpretation of statistically significant subsystem structures (*i.e.*, meso-level structures formed within a complex system) and the use of such structures for STS engineering and design. The findings presented herein shed light on the importance of treating significant subsystems as crucial functional units and building blocks of STSs and underscore the need to consider them in both macro-level system design and micro-level individual entity design for optimizing system performance and entity functionality. Beyond enriching systems science from the meso-level subsystem perspective, this dissertation is expected to generate broader impacts in: 1) Addressing imbalanced source allocations in societal infrastructure systems, such as uneven distribution of public resources in urban areas. By treating local communities as meso-level subsystems and utilizing their information, this research offers policymakers actionable insights for more efficient resource distribution; 2) showing the potential to inform robust design strategies for large networked physical systems like power grids and transportation networks, the meso-level subsystem-based approach facilitates the identification of critical functional units within these systems. Subsequently, system optimization design can be guided by preserving the functionality of these identified subsystems. 3) enhancing interdisciplinary collaboration between engineering and social sciences. The frameworks proposed in this dissertation are extensible to incorporate societal analytical models. For example, in the case study of customer-product market systems, a more advanced network model that integrates customer social networks into the proposed product competition network can be easily generated to support a more in-depth analysis. By bridging the gap between technical systems engineering and social aspects, it fosters a holistic approach to addressing complex societal challenges.

Table of Contents

[1]The content of this chapter has been published in (Xiao and Sha, 2022). My contributions include conceptualization, methodology, formal analysis, and article writing.

[2]The content of this chapter has been published in (Xiao et al., 2023a). My contributions include conceptualization, methodology, formal analysis, and article writing.

[3]The content of Section 5.2 has been published in (Xiao et al., 2024). My contributions include conceptualization, methodology, formal analysis, visualization, and article writing. The content of Section 5.3 has been published in (Gavino et al., 2023). My contributions include conceptualization, methodology, and article writing.

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Research Background and Motivation

The notion of *socio-technical* was originally proposed by (Trist and Bamforth, 1951) in the realm of labor studies, advocating for the integration of both knowledge accumulation and the enhancement of work environments within research endeavors. This approach, when applied within the discipline of systems engineering, gives rise to the concept of a *socio-technical system* (*STS*). The STS, an extension of socio-technical principles and conventional complex systems, embodies a framework where social and technical elements intertwine. An example of an STS is the customer-product market system. Customers' considerations and choices, playing as the social aspects, determine the market shares of a technical product; in turn, the design and development of products, being the technical aspects, affect customers' behaviors, and even interactions on social media. Other noteworthy examples of STS encompass sharing economy platforms, smart transportation systems, and the broader spectrum of emerging artificial intelligence (AI)-enabled systems (Heydari et al., 2020).

To better capture the inherent complexity of STS, complex networks have been proven to be a powerful representation for researching such systems. Its efficacy has been demonstrated across a spectrum of systems engineering applications. For example, Sha et al. (Sha et al., 2019b; Sha and Panchal, 2013a,b, 2016) conducted extensive research on network-based engineering design of complex systems, including the autonomous system-level Internet and the U.S. domestic air transportation systems. In addition, a series of studies have been conducted on the application of stochastic network models (*e.g.*, the Exponential Random Graph Model) to investigate customer-product interactions in vehicle market systems (Wang et al., 2016b; Fu et al., 2017; Wang et al., 2016a; Sha et al., 2018; Bi et al., 2018; Wang et al., 2018; Sha et al., 2019a; Cui et al., 2020). An additional advantage of network modeling lies in its ability to furnish a diverse array of statistical descriptors, facilitating

the characterization of individual components, their interactions, and the network holistically. Representative descriptors encompass degree centrality (quantifying the number of connections an individual possesses), network density (reflecting the observed connections relative to the total possible connections within a network), and average clustering coefficient (indicative of the tendency for individual components to form clusters) (Barabási and PÃ3sfai, 2016).

In addition to system representations, the complexity of STS (*e.g.*, large scale, complex interactions, and evolutionary dynamics) is another aspect casting new challenges when applying existing systems engineering and design methodologies to STS. First, it makes traditional top-down design approaches (*e.g.*, the waterfall model) (Crespi et al., 2008; Kramer, 2018) inapplicable to support STS engineering and design decisions. In a top-down design framework, designers often begin by identifying the overall specification and requirements of the system through customer analyses, which define the design space and design constraints (Takai et al., 2011). However, such a process cannot be applied to STS engineering and design due to spontaneous interconnections between individual entities (including humans) and undetermined system scale (*e.g.*, dynamic expansion and contraction often occur in most STS). Second, existing bottom-up design approaches believe that systems evolve as a result of interactions among individual entities (Trinh and Sha, 2022) and seek to gain deep insights into the process by which individual actions affect system-level performance (Sha, 2015), as illustrated in Figure 1.1. Although these approaches can address the issues posed by dynamic changes and complex interactions in STS, they oversimplify the analysis of STS structure, performance, and evolution by solely analyzing the behavior and interactions of individual entities, while ignoring the role and functionalities of subsystems.

However, subsystems have been proven to have significant impacts on system performance across various domains. Representative examples and their interpretations are introduced in Table 1.1. In social networks, researchers have analyzed

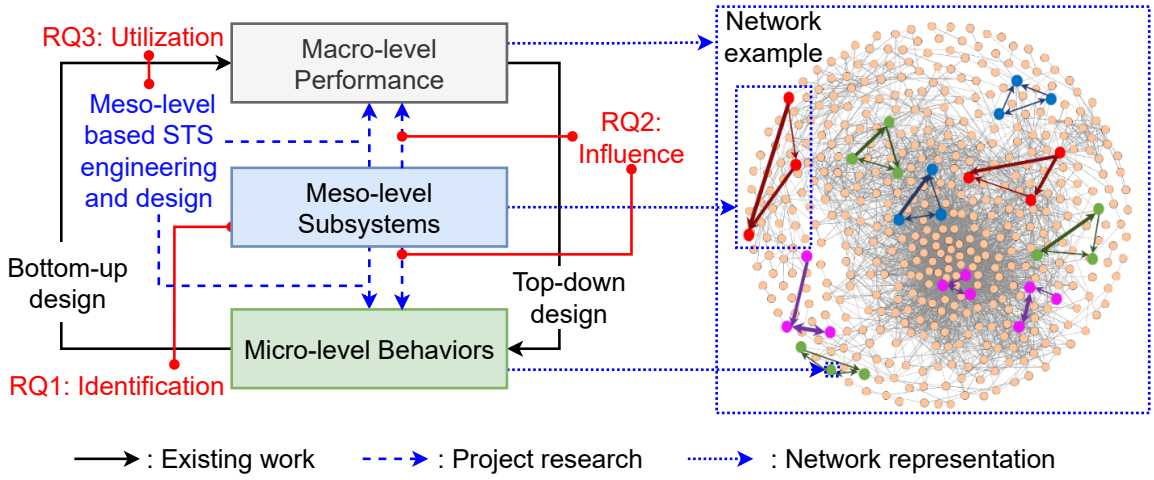
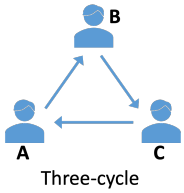
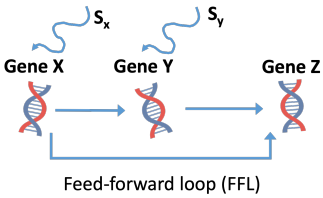
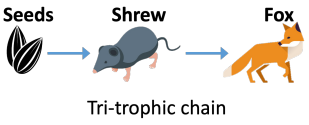
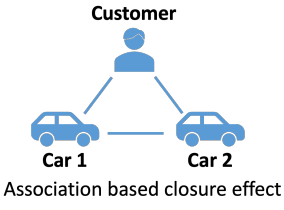


Figure 1.1: Network-based three-level decomposition of STS and associated research questions.

the triad census, recognizing it as the fundamental unit of social relations (Wasserman and Faust, 1994; Ribeiro et al., 2021). In biological systems, the feed-forward loop has been identified as a critical functional pattern prevalent in various biological networks, including metabolic networks (Alon, 2007). Ecologically, patterns such as the tri-trophic chain, exploitative competition, and omnivory characterize interactions within regional food webs, shedding light on community assembly and the driving forces behind local food web formation, such as predation and resource limitation (Baiser et al., 2016). Moreover, network motif theory has been utilized in computer science to develop higher-order networks (Rossi et al., 2018; Shao et al., 2021)leading to more accurate node embeddings. Furthermore, in the realm of socio-technical systems, researchers have identified significant interaction patterns between social actors and technical elements (Manny, 2023). For instance, the socio-technical triad observed in an urban wastewater system involves sensor data transferring to two workers responsible for the same pumping station. It has been investigated to understand its implications for infrastructure management (Manny et al., 2022). Similarly, the association-based closure effect in market systems, detailed in Table 1.1, has been explored to discern its influence on customer product purchase and consideration decision-making (Wang et al., 2016a).

Table 1.1: Examples of subsystems across different domains.

Discipline	**Significant Subsystem Structure**	**Interpretation**
Social Network		The *three-cycle effect* in friendship networks is often interpreted as an indication of the presence of local hierarchies in friendship networks. This implies that unreciprocated friendships may reflect dominance relations, where the recipient of a tie is perceived to occupy a higher position than the sender (Block, 2015).
Gene Transcription Network		The *Feed-forward loop* comprises transcription factors X and Y, which jointly regulate the transcription rate of target gene Z by binding to its regulatory region. Additionally, the FFL is influenced by two input signals, inducers Sx and Sy, which can activate or inhibit the transcriptional activity of X and Y (Mangan and Alon, 2003).
Food Web		In the context of food web networks, the *Tri-trophic chain* represents a specific pattern of interactions among three species at different trophic levels, typically consisting of a predator, a prey, and a resource species. It highlights the dynamics of predator-prey relationships within ecological communities (Baiser et al., 2016).
Market System		*The association based closure effect* indicates the likelihood of a closed structure involving two product nodes connected by an association link. For instance, it investigates whether customers are less inclined to consider two cars with many shared features simultaneously or not (Wang et al., 2016a).

In summary, despite the fact that recent efforts have tremendously enriched STS engineering and design, fundamental obstacles posed by the complexity of STS remain. A major reason is that there is a fundamental **knowledge gap** to understand what meaningful subsystem information is and how it can be extracted and used to guide the design of STS for desired system-level performance. To fill this knowledge gap, my dissertation research decomposed STS into three levels and adopted complex network theory as the foundation for investigation *i.e.*, macro-level performance, meso-level subsystems, and micro-level behaviors, as shown in Figure 1.1. Taking the shared mobility system as an example, the macro-level performance depicts the aggregated satisfaction of users with the overall service provided by the entire system; the meso-level subsystems represent significant travel patterns emerged between stations; the micro-level behaviors indicate single stations providing bike rental and return services to users. Then, on the basis of such a decomposition, my dissertation developed a meso-level network-based STS engineering and design approach to address the knowledge gap in three aspects: 1) validating the existence of the significant meso-level subsystems in an STS; 2) demonstrating the impacts of the critical meso-level systems on the macro-level performance and the micro-level behaviors; 3) proving the utilizability of the meso-level information to guide STS engineering and design.

1.2 Research Overview

The **research objective** of my dissertation is to develop a novel meso-level network-based framework for STS engineering and design. The **central hypothesis** is that in STS, subsystems at the meso level (*e.g.*, small clusters of individual entities) serve as critical links in system structures and could influence both macro-level performance and micro-level interactions, and thus deserve scientific investigation in STS design. The objective will be achieved by answering the following three **research questions**.

- ***RQ1:*** *How can significant meso-level system structures be identified?*

- ***RQ2:*** *What are the influences of the significant meso-level subsystems on the system performance at the macro level and the interaction mechanism at the micro level?*

- ***RQ3:*** *How can meso-level structural information be used to design an STS to achieve desired macro-level performance and micro-level functionality?*

Table 1.2 summarizes the central research objective, hypothesis, and research approaches. Accordingly, an overview of the proposed meso-level network-based design framework integrating all the research approaches step-by-step is given in Figure 1.2. The proposed framework entails six major steps. **Step One** is to address the data availability issue that is common in data-driven research given potential reasons such as commercial values of the market system data and the time-consuming process of data collection. In this work, the information retrieval and survey design approaches are implemented to address this issue. In **Step Two**, the objective is to develop a complex network model to represent the target STS including node and link definition as well as their attribute identification. Note that a good representation model requires a comprehensive understanding of the domain to which the STS belongs, such as knowledge of its design parameters and uncertain noise effects. Both **Step One** and **Step Two** serve as the foundation for the steps that follow.

Step Three, corresponding to ***RQ1***, is about significant sub-network mining using technologies such as network motif theory (introduced in Section 2.4.1) and exponential random graph model (ERGM) (introduced in Section 2.4.2). Once the sub-networks are obtained, descriptive analyses are conducted for interpretation. Next, the identified meso-level subsystems are treated as inputs of **Step Foure**. In **Step Foure**, the objective is to determine the quantitative influences of those significant meso-level subsystems on the system performance at macro level and individual functionality at micro level, as well as use that information to inform system engineering and design decisions. This step helps reveal the answers to ***RQ2*** and ***RQ3***

and is split into multiple subtasks. The first subtask is to define meso-level network-based criteria for evaluating system performances at macro, meso, and micro levels and depicting their interplays. Secondly, the design problem of interest is described using the proposed performance criteria, such as transforming shared mobility system robustness design to an optimization of the system's capacity planning decisions. Finally, the design problem is solved, and the implementation of the achieved design solution is represented by the update to the complex network model. Since a new design strategy for large-scale STS is hard to test in the real world because of the cost and risks of failures, a predictive model is required to test and validate the effectiveness of the new design method in a short period with little cost. Therefore, **Step Five** of the proposed framework is to develop an STS predictive model by methods such as ERGM and graph neural network (GNN) (introduced in Section 2.4.3). The merit of using GNN for prediction is that because the GNN model takes network neighborhood information into account, the importance of meso-level network information can be assessed by evaluating its contribution to improving prediction accuracy, thereby bringing insights into ***RQ2***. In **Step Six**, the predicted system performance after implementing the optimal design solution is evaluated to assess its reliability, addressing ***RQ3***.

Lastly, this dissertation initiates a preliminary investigation into meso-level temporal subsystem-based analysis of socio-technical systems (STS) dynamics. The initial focus lies on addressing the scarcity of STS dynamic data. Subsequently, the methodologies outlined in **Step Two** and **Step Three** are enhanced to incorporate temporal dinemsions, notably by incorporating time-related attributes into the network model (*e.g.*, temporal features assigned to nodes and links). Ultimately, this study delves into the examination and identification of key characteristics of STS dynamics and significant temporal subsystems, employing these refined approaches.

Table 1.2: Table showing central research objective, hypothesis, and research approaches.

Central Research Objective	To develop a *meso-level network-based framework* for complex socio-technical systems (STS) engineering and design
Central Hypothesis	In STS, subsystems at the meso level (*e.g.*, small clusters of individual entities) serve as critical links in system structures and could influence both macro-level performance and micro-level interactions.
Research Approaches	The *meso-level network-based design framework* for STS engineering and design: • **STS data collection:** collecting STS data through information retrieval and survey design. • **STS network representation:** representing STS with a complex network model, interpreting design parameters and noise effects. • **Significant subnetwork mining:** identifying significant subsystems of STS using network-motif theory or network-based statistical inference models (*e.g.*, exponential random graph models (ERGMs)). • **STS optimization design:** developing an optimal STS design approach by formulating and solving a subsystems-based optimization model. • **STS prediction:** building a predictive model (graph neural network (GNN) or ERGM-based) to support STS design decisions and validation. • **Design solution evaluation:** Implementing the STS predictive model to assess the reliability of the optimal design solution by predicting system performance after design updates.

1.3 Validation Cases

This dissertation employs two applications to validate the approaches outlined in Table 1.2. The first application domain pertains to large service systems, while the second domain relates to socioeconomic systems. Specifically, the shared mobility system serves as an illustrative example in the domain of large service systems. Meanwhile, the customer-product market system is chosen for illustration within the realm of socioeconomic systems. The rationale for selecting cases from different domains is twofold: Firstly, it aims to validate the generality of the proposed models,

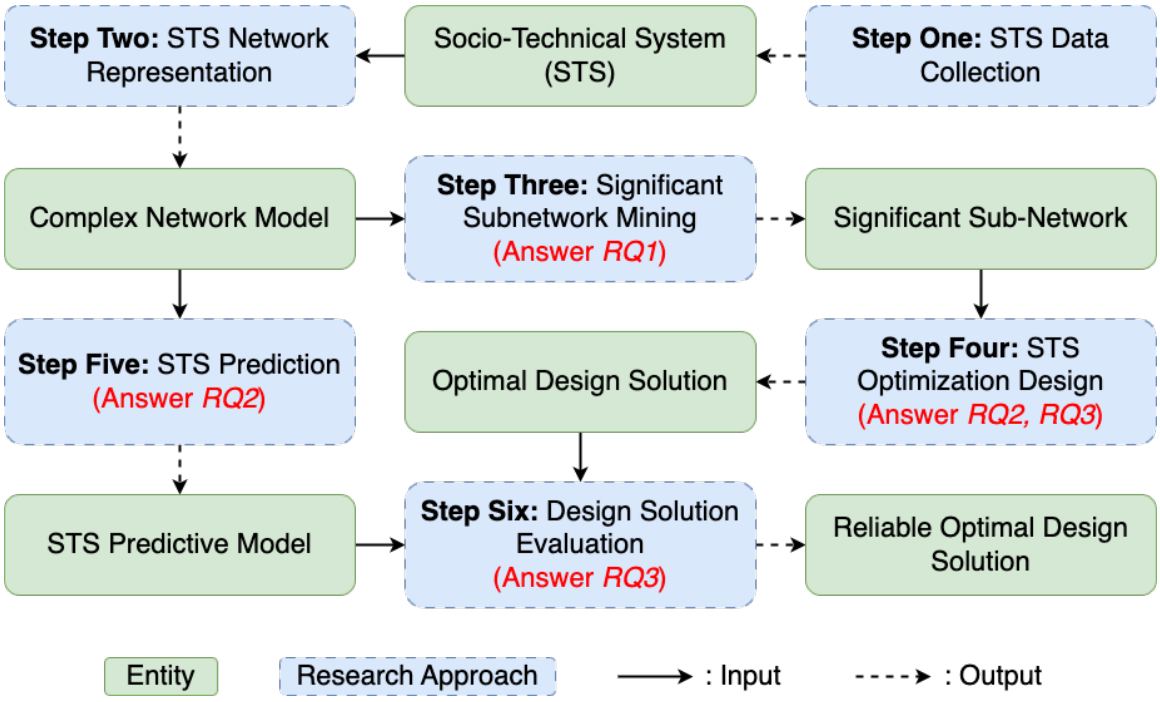


Figure 1.2: Meso-level network-based design framework for STS engineering and design.

approaches, and computational techniques introduced in the research. Secondly, the shared mobility system is utilized to showcase the effectiveness of the proposed meso-level network-based design approach in enhancing macro-level system performance (*e.g.*, system-level user satisfaction), which is typically the focal point for system stakeholders. Conversely, the customer-product market system is employed to demonstrate the effectiveness of the proposed meso-level network-based design approach in enhancing the functionality of individual entities at the micro-level (*e.g.*, the popularity of a single product in the market), which holds significance for corresponding market players.

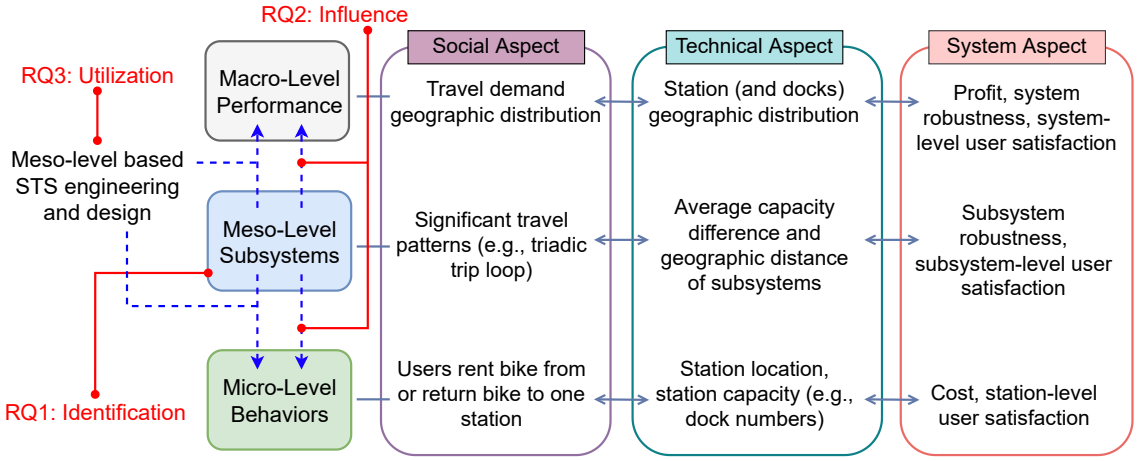
Figure 1.3 shows the mapping of each case study onto micro-level, meso-level, and macro-level, along with associated research questions. Table 1.3 illustrates the detailed research tasks associated with each case. The research tasks pertaining to shared mobility systems (SMS) encompass: 1) constructing an SMS network representation to simulate user travel behaviors within the system and employing significant

subnetwork mining techniques to identify significant user travel patterns within subsystems (local service systems comprising multiple stations); 2) quantifying the impacts of the identified travel patterns on the macro-level system robustness to against seasonal fluctuations in usage demand; 3) formulating and solving an optimization model based on subsystems to enhance the seasonal robustness of the SMS; 4) developing a predictive model that incorporates meso-level system information to forecast travel demands between stations; 5) establishing a design decision support framework to evaluate macro-level SMS design decisions, utilizing the developed predictive model.

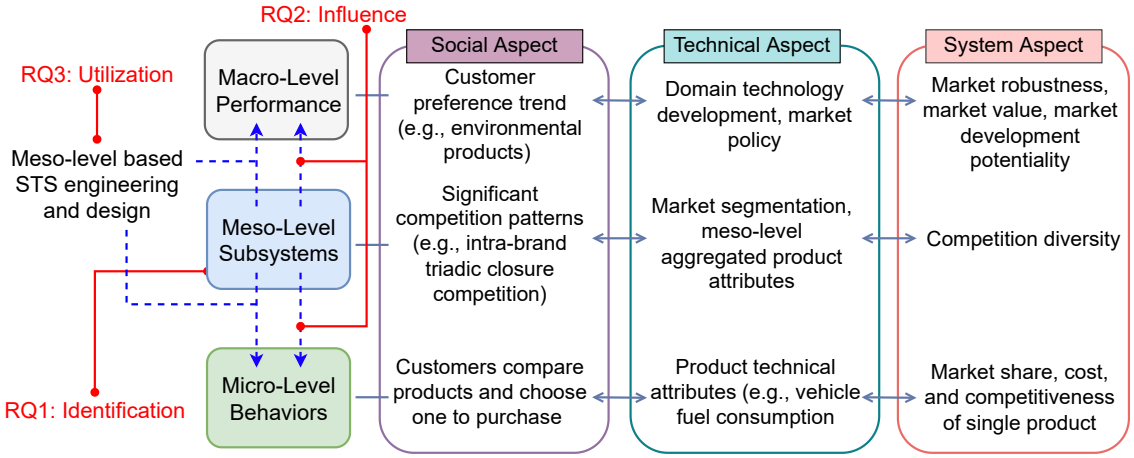
Regarding the research tasks of customer-product market system (CPMS), they include: 1) implementing the information retrieval and survey design approach to gather network data pertaining to the CPMS, including product attribute data and relational data among products; 2) constructing a CPMS network representation to model the competitive relationships among products, and utilizing significant subnetwork mining techniques to identify significant product competition patterns; 3) quantifying the competitiveness of products within the market based on the identified patterns of competition; 4) formulating and solving an optimization model grounded in the derived quantification to improve the competitiveness of individual products. Lastly, both SMS and CPMS a employed to preliminarily investigate the meso-level temporal subsystem-based analysis of socio-technical systems (STS) dynamics. The tasks include: 1) exploring the potential solutions for the lack of STS dynamic data; 2) defining temporal network representation to model the evolution of STS and developing corresponding significant subnetwork mining approaches to identify significant temporal subsystems.

1.4 Contributions

This dissertation centers on the engineering and design of complex socio-technical systems (STSs), with a primary focus on elucidating the influence of meso-level sub-



(a) Shared mobility system



(b) Customer-product market system

Figure 1.3: Three-level decomposition of validation cases and associated research questions

systems on macro-level system performance and micro-level individual entity functionality. Its **primary contribution** lies in the development of a meso-level network-based framework for STS engineering and design. This framework encompasses various components, including network modeling, identification of significant subsystems, optimization design based on identified subsystems, development of predictive models for STS, and evaluation of design solutions based on predictive modeling. The specific contributions of this dissertation are:

Table 1.3: Summary of research tasks associated with the case studies.

Case Study	Research Tasks
Shared Mobility System (SMS)	• Implement the STS network representation and significant subnetwork mining approaches to build complex network models and identify significant subsystems. • Validate the quantitative influences of those identified crucial subgraphs on macro-level system performance by examining the relationship between the subgraph features (either technical design or network structures features) and the entire system operation. • Validate the meso-level network-based STS optimization design approach for the optimization of macro-level STS design decisions. • Develop a predictive model considering meso-level system information to demonstrate its ability to forecast macro-level system operations and micro-level individual behaviors. • Validate the STS design decision support framework for assessing macro-level STS design decisions using the developed predictive model.
Customer-Product Market System (CPMS)	• Implement the information retrieval and survey design approach to collect customer-product market data. • Implement the STS network representation and significant subnetwork mining approaches to build complex network models and identify significant subsystems. • Validate the quantitative influences of those identified crucial subgraphs on micro-level entity mechanism by examining the relationship between the subgraph features (either technical design or network structures features) and individual entity mechanism. • Validate the meso-level network-based STS optimization design approach for the optimization of micro-level entity design decisions.
SMS & CPMS	• Explore the potential solutions for the lack of STS dynamic data. • Implement the STS network representation and significant subnetwork mining approaches to build temporal network models and identify significant temporal subsystems.

1. **An information retrieval and survey design framework for networked socio-technical system data collection.**

   Within this framework, our contributions encompass the collection of two types

of product attribute data: US household vacuum cleaners and US vehicle models from the years 2016 to 2022, facilitated through web crawling techniques. Furthermore, we have developed a web-based survey platform that facilitates interactive information retrieval and virtual online shopping, enabling the collection of data on product co-consideration relationships. Lastly, we have contributed to the extraction of co-mentioning relationship data from social media text data using natural language models. These methods for collecting entity attribute data and corresponding relationship data constitute the cornerstone of our STS engineering and design studies, addressing significant data scarcity challenges. To foster broader scholarly engagement, both the entity and relationship datasets will be made publicly available, thereby facilitating research endeavors focused on STS design studies.

2. **A framework for macro-level STS robust design against seasonal effects by capacity planning decisions optimization.**

   In this framework, we introduced the network modeling of STS and the significant subnetwork mining approaches. Then, we contributed to proposing three subsystem-based metrics for system performance evaluation and capacity planning decision-making. The first one is the usage demand imbalance score of a local service system, the second one is the measurement of a subsystem's seasonal robustness, and the third one is a capacity planning decision criterion. Based on these three metrics, we validate that the sensitivity of STS performance against seasonal effects is highly correlated with the imbalanced capacity between service nodes in an STS. Correspondingly, we formulate a design optimization problem to improve the robustness of STS by rebalancing the resources at critical service nodes. The development of this framework helps with addressing ***RQ1***, ***RQ2***, and ***RQ3***.

3. **A complex network-based prediction framework for STS design support with graph neural network.**

In this framework, we have made significant contributions by proposing a complex network-based approach founded on graph neural network (GNN) techniques for predicting links between individual entities within STSs. By contrasting with conventional artificial neural network (ANN) models, we have demonstrated that the inclusion of meso-level subsystem information enhances the predictive performance of the model by 8%. Additionally, we have examined the performance of our proposed predictive model under varying link strengths, ranging from weak to strong. Our findings consistently indicate that the predictive model incorporating subsystem information consistently outperforms the ANN model lacking such information across different network densities and typologies. Furthermore, our innovative approach involves linking STS design decisions with the network link prediction problem. This linkage provides system designers with a tool to test and experiment with their design strategies, which is particularly crucial in the realm of complex STS research where the verification and validation of system designs pose significant challenges. The development of this framework contributes significantly to addressing ***RQ2*** and ***RQ3***, thereby advancing the understanding and practice of STS engineering and design.

4. **A micro-level entity design framework considering meso-level dependencies.**

   This framework creatively used network representations to characterize and capture dependencies and relations between individual entities in STS and integrate these representations into design formulations to find optimal decisions for the desired performance of a system. Specifically, the framework entails the following steps: 1) mining significant local networks, 2) transforming the original system design objective using the identified local networks, and 3) searching optimal design attributes for desired system performance by combining the genetic algorithm and a network predictive model. The development of this framework

helps with addressing ***RQ1***, ***RQ2***, and ***RQ3***.

5. **A meso-level temporal subsystem-based analyzing framework of STS dynamics.**

   In this framework, our initial contribution involves a comprehensive exploration of utilizing a GNN-based link prediction model, which integrates data undersampling methods and model post-processing techniques. This amalgamation serves to mitigate the challenge of dynamic data scarcity in socio-technical systems (STSs). Subsequently, we introduce a method for analyzing meso-level temporal subsystems. This approach encompasses temporal network modeling, significant temporal subsystem mining, and the empirical interpretation of corresponding temporal characteristics. The development of this framework is instrumental in addressing ***RQ1*** and ***RQ2***, thereby enhancing our understanding and analytical capabilities of the temporal dimension in STS engineering and design.

1.5 Overview of the Dissertation

The dissertation is structured into eight chapters, with an overview and roadmap provided in Figure 1.4. Chapter 2 conducts a literature review on two application examples, networked large service systems and product market systems, identifying research gaps in existing literature. Additionally, it introduces the technical background associated with network motif theory, the exponential random graph model (ERGM), and the graph neural network (GNN) model. Chapters 3 and 4 present the macro-level STS seasonal robust design framework and the network-based prediction framework. The objective of these chapters is to evaluate system performance and conduct optimization design at the macro level, with a focus on utilizing meso-level subsystem support. Case studies involving the shared mobility system (SMS) are employed for validation. Chapters 5 and 6 introduce the information retrieval and survey design framework and the micro-level entity design framework. These

chapters aim to evaluate individual entity functionality and conduct optimal design at the micro level, supported by meso-level subsystems. Case studies involving the customer-product market system (CPMS) are utilized for validation. In Chapter 7, the meso-level subsystem-based STS dynamic analysis is introduced. This includes exploring methods to address dynamic data scarcity issues, dynamic network modeling, and significant temporal subnetwork mining. Case studies involving both SMS and CPMS are employed in this chapter. Finally, Chapter 8 summarizes the contributions of this dissertation and suggests areas for future research.

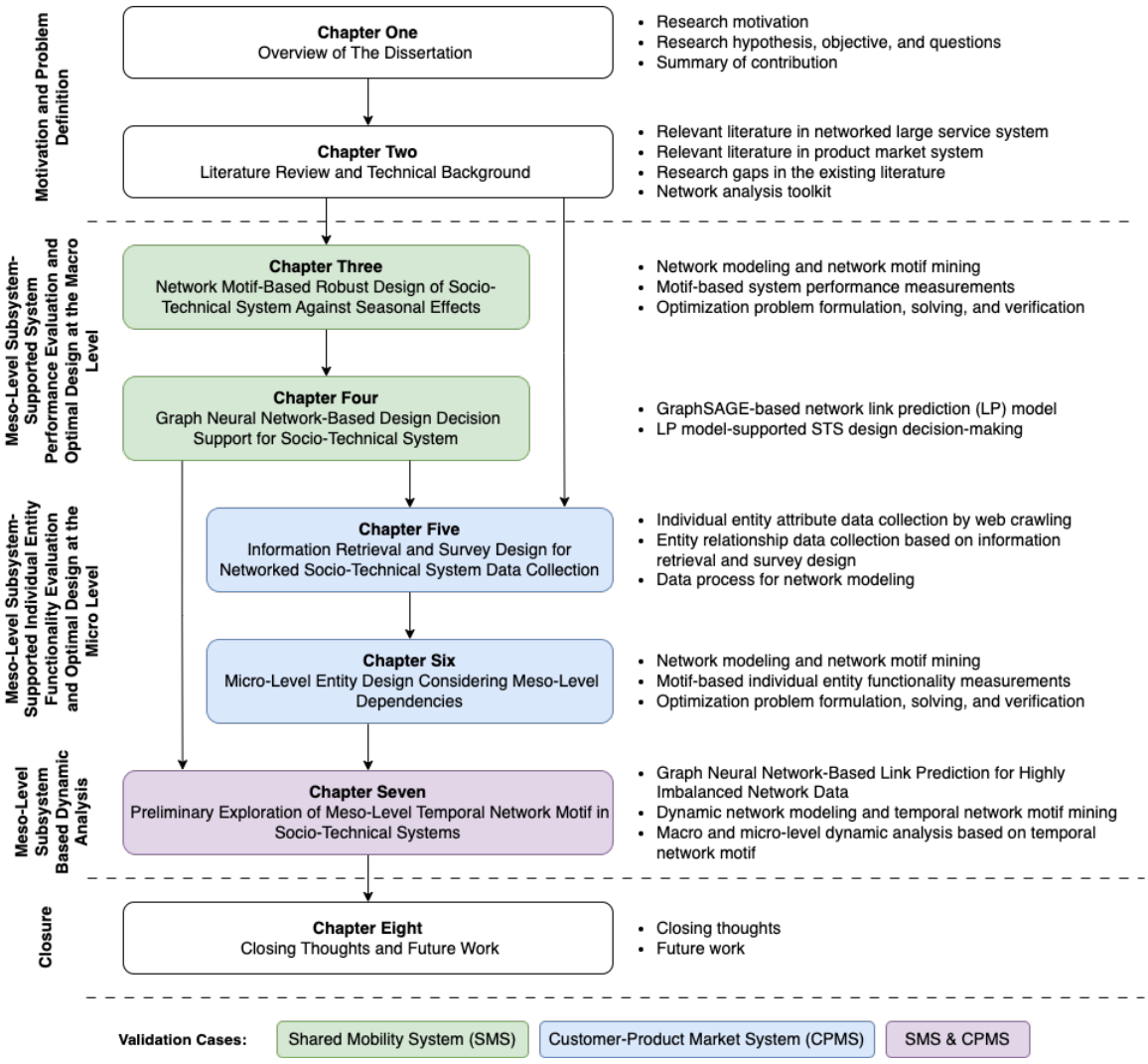


Figure 1.4: Dissertation Roadmap.

Chapter 2: Literature Review and Technical Background

This dissertation employs two application examples to validate the proposed framework and approaches. The first application pertains to networked large service systems, such as shared mobility systems, power grids, and airline systems. The second application focuses on product market systems. The aim of this chapter is to introduce these two applications and identify research gaps in the existing literature. The chapter is structured into three sections. The first two sections correspond to these two applications, providing detailed insights into their characteristics and challenges. Table 2.1 summarizes the structure of the literature review associated with these two applications and highlights the corresponding research gaps. The last section introduces the network analysis toolkit, which includes network motif theory, exponential random graph model (ERGM), and graph neural network (GNN). These components form the technical background of this dissertation.

2.1 Networked Large Service Systems

Networked large service systems play a crucial role in modern societies, encompassing a wide array of domains such as transportation, telecommunications, energy distribution, and healthcare (Bai et al., 2014; Shaqsi et al., 2020; Fjeldstad et al., 2020; Keyhani, 2016). These systems are integral to providing essential services to individuals and organizations, facilitating communication, transportation, energy supply, and various other functions. As defined in the literature, a service system is described as "a composite of agents, technology, environment, and/or organization units of agents and/or technology, functioning in space-time and cyberspace for a given period of time" (Stanicek and Winkler, 2010). Service systems belong to the broader category of socio-technical systems (STSs), characterized by the interplay between

Table 2.1: Mapping of literature review and research gaps.

Literature Review	**Research Gaps**
Existing studies on networked large service systems	• Knowledge foundation for the rationale of meso-level subsystems' impact on macro-level service system robustness is lacking. • A validation framework by developing a predictive model to support service system design decisions assessment is lacking. • Knowledge foundation for the rationale of the significant temporal subsystems in large service systems is lacking.
Existing studies on product market systems	• An efficient framework for product market data collection is lacking. • A product optimization design framework considering meso-level significant competition patterns is lacking. • An efficient dynamic competitiveness quantification of a product in a market is lacking.

social and technical components. The technical aspects of service systems encompass the services provided, such as telecommunications or electricity supply, as well as the technical infrastructure supporting these services. On the other hand, the social components include various stakeholders involved in the system, such as customers/users, operators, and regulators (Li et al., 2020; Wang, 2013). In essence, service systems operate within a networked structure, where interactions and dependencies between different components are prevalent (Wang, 2013).

In this dissertation, we reviewed the existing efforts in service system engineering and design from a social-technical perspective and the ones based on complex network science. These efforts have focused on various aspects aimed at enhancing system performance, resilience, and efficiency. some representative works in earlier stages include: 1) a discussion about socio-technical systems thinking as a tool for the concurrent development of organizational business models and associated service offerings that deliver value provided to customers and suppliers (Beaumont et al., 2014); 2) a comprehensive analysis of the rail and bus transportation networks in

Singapore, focusing on their topological and dynamical properties. Significant differences in edge weights were observed between weekdays and weekends, highlighting the importance of considering temporal effects in network analysis (Soh et al., 2010); 3) a comprehensive summarization of the application of complex network analysis (CNA) techniques to study the properties of electricity distribution power grid infrastructures, which highlights the importance of understanding the power grid as a complex system due to its societal relevance and impact on infrastructure reliability (Pagani and Aiello, 2013).

More recently, given that the service systems are on a larger scale than ever before, including more service agents that hugely increase the spatial-temporal operation complexity, their engineering and design poses new challenges for seeking more computationally efficient methods. In addition, the increasing complexity introduced by the unpredictable features of the social aspects runs through the three broad stages in the system engineering lifecycle: analysis, design, and evaluation (ElMaraghy et al., 2012; Baxter and Sommerville, 2011), and continues to affect the functionality of the entire system. It is also this complexity that raises a high requirement of the system robustness against various disturbances (Kalsi et al., 1999; Gribble, 2001). To address these challenges, De Bona *et al.*introduced a reduced model for analyzing public transportation networks (PTNs) by preserving network skeleton through the removal of 2-degree nodes, revealing hidden network structures and characteristics (De Bona et al., 2021). In another interdisciplinary study, drawing inspiration from the structure and functioning of biological ecosystems that have survived disturbances over millions of years, Rodriguez *et al.*explored the decentralization of water storage tanks in urban water networks, finding that such decentralization significantly improves network resilience and the system's ability to meet demand during disruptions while reducing freshwater consumption (Rodriguez et al., 2023). With the advancement of deep learning methodologies, Wu and Wang introduced a generative design method utilizing graph learning algorithms, enabling the efficient creation of resilient system designs by mining good properties from existing systems and predicting performance

for iterative improvement (Wu and Wang, 2023).

Back to the disturbances causing system robustness issues, one typical disturbance that can impact numerous service systems across domains is seasonal effects(Markolf et al., 2019; Sun et al., 2015; Xie et al., 2021; Gabrielli et al., 2019). Taking the shared mobility system as an example, seasonal effects not only require the systematic design of station distribution and the capacity of each station to fight against varied weather conditions but could also generate demand fluctuation in different months that affects system operational performance. The representative studies that aim to improve the service systems' seasonal robustness include: 1) a comprehensive exploration of direct and indirect pathways of disruptions that lead to the vulnerability of the transportation systems. Direct disruption pathways involve abrupt impacts on physical infrastructure, alongside effects stemming from non-physical factors like human health, behavior, and decision-making. Similarly, indirect disruption pathways arise from interconnections with critical infrastructure and social systems (Markolf et al., 2019); 2) a framework for the robust design of multi-energy systems with limited input data, aiming to minimize total annual costs and CO2 emissions through uncertainty analysis. It addresses the optimal design of decentralized systems involving renewable energy sources and energy storage technologies, evaluating system performance under different scenarios to define a robust scenario for design purposes (Gabrielli et al., 2019).

Despite the existing efforts that stress the equal importance of both technical and societal aspects in service systems, propose complex network-based service system modeling and descriptive analysis, and integrate interdisciplinary knowledge and advanced deep learning algorithms, there are limited studies focusing on understanding the essential impact of meso-level significant subsystems on macro-level system performance. These subsystems typically function as the units of functionality within socio-technical systems (STS), as discussed in Chapter 1. Only works by Manny *et al.*(Manny et al., 2022; Manny, 2023) have proposed a socio-technical network perspective to study infrastructure systems and introduced socio-technical

motifs embedded in these systems. However, these studies are limited to analytical methods for studying basic network concepts and empirical analysis of socio-technical motif functionality, lacking a knowledge foundation for understanding the quantitative impact of meso-level subsystems on macro-level service system robustness. Similarly, there is a need for a quantification method to assess the impact of temporal subsystems on large service systems. Lastly, existing studies lack a validation framework involving the development of a predictive model to support service system design decision assessment.

2.2 Product Market Systems

The product market system represents a significant type of socio-technical system pivotal to modern economies, facilitating goods exchange, stimulating innovation and competition, and shaping consumer behavior (Rosa et al., 1999). Its social aspects encompass a diverse array of stakeholders, including customers, market players, and policymakers. Conversely, its technical facets encompass product attributes, advertising mechanisms, and market-regulating policies, among others. This dissertation adopts a market players' perspective, with the aim of designing products that cater to customer preferences, a critical determinant of success in competitive markets. Given this context, companies are keenly interested in understanding the factors influencing customer purchasing behaviors and their relative importance. Over the past decades, customer preference modeling has emerged as a primary research method to address these questions in both marketing science (Stankevich, 2017; Pescher and Spann, 2014) and the engineering design community. For instance, the practice of customer preference modeling offers designers valuable insights into discerning preferred product features and understanding how customers prioritize attributes (Pescher and Spann, 2014; Sha et al., 2017). Additionally, scholarly investigations indicate that customer decision-making typically unfolds in two stages: the formation of a consideration set followed by the final selection based on different criteria (Shocker et al.,

1991). Research interest in customer preference modeling has predominantly revolved around two key areas. Firstly, scholars have sought to unravel the influence of product attributes on customer decision-making processes, employing techniques such as customer-product network modeling to analyze how design attributes impact customer considerations and choices (Cui et al., 2020; Bi et al., 2021; Sha et al., 2017; Wang et al., 2015). Secondly, there has been a growing interest in understanding the role of social influence in customers' decision-making, exemplified by studies examining changes in customer-preferred attributes post-peer effects and the influence of demographic data from customers' social networks (Argo, 2020; Aral and Walker, 2011; Campbell and Lee, 1991).

Data scarcity issue However, a significant gap exists in the current literature, stemming from the separate investigation of the influence of social factors and product attributes on customer purchase decisions. This segregation is primarily due to data limitations in two aspects. Firstly, the absence of simultaneous collection of customers' social network data and attribute data for their considered and purchased products makes the creation of synthetic social network data when examining social influence on customer choices (Wang et al., 2016a). Secondly, many datasets originate from private sectors, where the inclusion of customer preferences makes them highly valuable to enterprises and thus unavailable for public sharing. Consequently, these constraints have hindered the reproducibility and repeatability of numerous existing models (Anon, 2013).To address these constraints, researchers are compelled to seek alternative data sources, such as online product reviews, social media platforms, and publicly available customer survey data. Online reviews, typically authored by purchasers, are accessible through e-commerce websites, offering valuable insights into product experiences (Lee and Bradlow, 2011). Social media data, comprising content shared by customers or experts on platforms like Twitter or YouTube, also provide valuable information (Tuarob and Tucker, 2015). However, both sources often lack comprehensive customer demographics, limiting their utility in customer

preference modeling. Public customer survey data, while offering insights into product preferences, often involves a limited selection of products, constraining modeling efforts (Bao et al., 2020; Barnard et al., 2016). Therefore, there is a need for an integrated systematic approach that merges information retrieval and survey design to facilitate data collection for customer preference modeling, thereby overcoming the aforementioned limitations.

Product market competition analysis The competitiveness of a company is the result of a combination of external and internal factors. External factors include 1) the inherent characteristics of a product market, such as its size associated with the volume of customer demand and market differentiation determined by diverse customer preferences; 2) its competitive environment shaped by all market participants and stakeholders. Internal factors involve a company's organizational forms, product strategies, and the speed of its response to changing technologies and market opportunities. For example, when a new technique is introduced, a competitive company can often rapidly master it to launch new products or upgrade existing products (Sanchez, 1995). To maintain a competitive position in the market in the long run, competition analysis is important for a firm to gain a thorough understanding of both the external and internal factors that influence its competitiveness. One external competition analysis example is to investigate the competitive environment of a market, such as studying customer preferences (Cui et al., 2020) of its representative products and typical competition patterns (*e.g.*, how products compete between brands). An internal competition analysis example is for a company to generate a better understanding of the market positions of its own products, such as the market share of the most popular product or the one that always competes against other brands.

In recent decades, competition analysis of product markets has received significant attention. In particular, researchers in the market science domain have contributed rich findings and analysis approaches (Sanchez, 1995; Spiegler, 2016). For example, Karuna determined the competition of a product market in three dimensions:

product substitutability, market size, and entry costs. Based on this determination, he demonstrated that companies offer stronger managerial incentives when industry competition is more intense (Karuna, 2007). In another instance, Bustamante and Donangelo explored the interrelation between the competitive environment in which firms operate and their exposure to systematic risk (Bustamante and Donangelo, 2017). More recently, researchers from engineering design communities utilized product market competition analysis to better understand the needs in engineering design. Wang et al. focused on product design for uncertain market systems (Wang et al., 2011). They proposed an agent-based approach to help firms make competitive product design and pricing decisions to face possible reactions from market players in the short and long runs. Yip et al. investigated the possibility of using a subset of competing products or composite products to replace a large set of competing products. They found that optimal product design decision is independent of any information about competitors when customer preferences are homogeneous, but this is not valid when customer preferences are heterogeneous (Yip et al., 2022). Wang and Chen et al. proposed using customer preference data to build competition networks. Then, various network-based competition analyses (*e.g.*, the evolution of product competitions) were generated, which were demonstrated using the vehicle market as case studies (Wang et al., 2018; Sha et al., 2018; Xie et al., 2020).

Recent efforts have significantly enhanced our understanding of how the market environment influences a company's operations. Notably, analyses of product market competition using network-based customer preference modeling have shed light on the existence and significance of meso-level competition patterns (Cui et al., 2020; Sha et al., 2018). However, a fundamental research gap persists, as the majority of these studies focus solely on solving forward problems. What is lacking in the existing literature is the solution to the inverse problem, which pertains to integrating the impact of identified competition patterns into the product design process to enhance market performance (*e.g.*, improve their market shares) Furthermore, there is a crucial knowledge gap concerning the dynamic quantification of a product's competi-

tiveness in the market. Such quantification is indispensable for companies seeking to adapt their strategies in response to evolving market dynamics, identify areas for improvement in product offerings, allocate resources effectively, and maintain or enhance their competitive edge. Therefore, addressing these gaps is imperative for companies aiming to thrive in competitive market environments.

2.3 Network Analysis Toolkit

This dissertation aims to create a novel local-level network-based framework for STS engineering and design, combining three advanced network theories and models, including network motif theory, exponential random graph model (ERGM), and graph neural network (GNN) model. This section provides a comprehensive introduction to the technical background of these three network theories and models.

2.3.1 Network Motif Theory

Network motifs are underlying nonrandom subgraphs within the complex networks. Before being named by (Milo et al., 2002), network motifs experienced a long research period (Stone et al., 2019), which was originally considered as certain patterns statistically emerging in real-world networks instead of the same-sized random networks (Holland and Leinhardt, 1974). Since then, motif research can be divided into two main subjects where the first one focuses on motif structure explanation (Alon, 2007; Felmlee et al., 2018; Paranjape et al., 2017), and the second one is keen on motif mining algorithms (Kashtan et al., 2004; Wernicke and Rasche, 2006; Choobdar et al., 2012). Motifs can be classified as directed or undirected and categorized by the number of nodes they comprise. Commonly studied motifs include size-2 motifs (dyads), size-3 motifs (triads), and size-4 motifs (tetrads) (Felmlee et al., 2018). Dyads, as the simplest motifs, play a crucial role in forming higher-level motifs and the entire network. Triads, also known as "transitivity" motifs, significantly influence the growth of social networks. Tetrads, a recent research focus, span various disciplines such as

biology, electronics, and social analysis. Given that the triad serves as the cornerstone of social relationships and is the most basic building block of many complex networks, size-3 motifs are selected as the focus of study for STS in this dissertation. Thus, only size-3 motifs are considered herein. The unique structures and IDs of size-3 directed motifs are illustrated in Table 2.2 (Rasche and Wernicke, 2006). For size-3 undirected motifs, there are only two unique structures sharing similarities with motif 238 and motif 78 in Table 2.2 and possessing the same adjacency matrices. Therefore, they are not reiterated here.

The identification of significant network motifs is conducted by comparing the real-world network $\boldsymbol{Y}_{obs}$ with a specific randomized network set[1] $\Omega(\boldsymbol{Y'})$ that includes N random networks. Only statistically significant local networks are considered as network motifs. The null hypothesis is that the frequencies of a local network in random networks $F_{rand}(\boldsymbol{y})$ are equal to or greater than that of the real-world network $F_{obs}(\boldsymbol{y})$. It is rejected if the *p-value* given in Equation (2.1) is less than a level of significance (commonly 0.01 or 0.05) (Schwöbbermeyer, 2008).

$$P(\boldsymbol{y}) = \frac{1}{N}\sum_{j=1}^{N}\delta(F_{rand}(\boldsymbol{y}) \geq F_{obs}(\boldsymbol{y})), \tag{2.1}$$

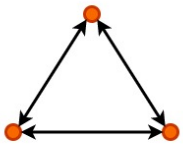
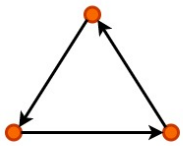
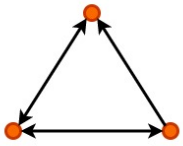
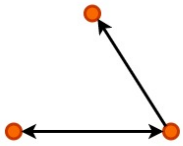
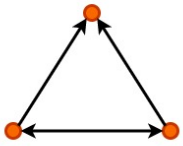
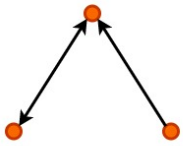
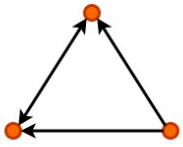
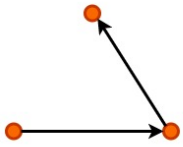
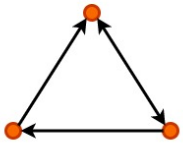
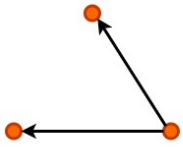
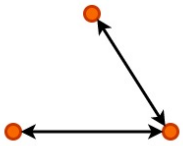
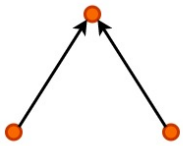
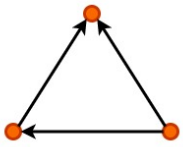
where δ is the sign function equal to 1 when $F_{random}(\boldsymbol{y}) \geq F_{obs}(\boldsymbol{y})$, and 0 otherwise. In addition, the *Z-score* is another measurement of the significance of a network motif, which is defined as

$$Z(\boldsymbol{y}) = \frac{F_{obs}(\boldsymbol{y}) - \mu_{rand}(\boldsymbol{y})}{\sigma_{rand}(\boldsymbol{y})}, \tag{2.2}$$

where $\mu_{rand}(\boldsymbol{y})$ and $\sigma_{rand}(\boldsymbol{y})$ represent the mean and standard deviation of $F_{rand}(\boldsymbol{y})$. A higher *Z-score* indicates that $\boldsymbol{y}$ is a more significant motif in $\boldsymbol{Y}_{obs}$ (Schwöbbermeyer,

[1]For a rigorous comparison, each node in the random networks has the same number of degrees as the corresponding real-world network. Moreover, the random networks used to calculate the significance of size-n local networks are generated to keep the same number of occurrences of all size-$(n-1)$ local networks as in the real-world network (Milo et al., 2002)

Table 2.2: Size-3 directed network motif list*.

ID	Structure	Adjacent Matrix	ID	Structure	Adjacent Matrix
238		0 1 1 1 0 1 1 1 0	140		0 1 0 0 0 1 1 0 0
174		0 1 0 1 0 1 1 1 0	14		0 0 0 0 0 1 1 1 0
46		0 0 0 1 0 1 1 1 0	164		0 1 0 1 0 0 1 0 0
166		0 1 0 1 0 0 1 1 0	12		0 0 0 0 0 1 1 0 0
102		0 0 1 1 0 0 1 1 0	6		0 0 0 0 0 0 1 1 0
78		0 0 1 0 0 1 1 1 0	36		0 0 0 1 0 0 1 0 0
38		0 0 0 1 0 0 1 1 0			

*: The motif IDs determined by (Rasche and Wernicke, 2006) consider each motif's adjacent matrix as a binary representation and transform the binary representation to a decimal number. For example, the binary representation of the decimal number 174 is 010101110, which is consistent with the adjacent matrix of motif 174. Regarding ordering the motifs in Table 2.2, from top to button and left to right, we rank them based on the number of their arrows from large to small.

2008).

2.3.2 Exponential Random Graph Model (ERGM)

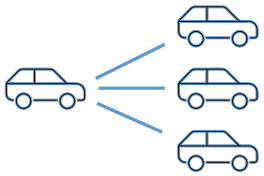
Exponential Random Graph Model (ERGM) is a stochastic network model for the observed network $\boldsymbol{Y}_{obs}$, representing a particular instance within a collection of potential random networks $\boldsymbol{Y}$. These random networks conform to the distribution described in Equation (2.3) (Robins et al., 2007; Sha et al., 2023).

$$Pr(\boldsymbol{Y} = \boldsymbol{Y_{obs}}) = \frac{exp(\boldsymbol{\theta}'\boldsymbol{g}(\boldsymbol{Y_{obs}}))}{\kappa(\boldsymbol{\theta})}, \tag{2.3}$$

where $\boldsymbol{\theta}$ is a vector of model parameters, and $\boldsymbol{g}(\boldsymbol{Y_{obs}})$ is a vector of the network statistics defining various network structures that can incorporate either nodal attributes or edge attributes. To ensure that Equation (2.3) constitutes a valid probability distribution, $\kappa(\boldsymbol{\theta})$ serves as a normalizing factor. Equation (2.3) suggests that the probability mass function on the network space is proportional to the exponential of a linear combination of network statistics. The formulation also indicates that the network with statistics in $\boldsymbol{g(Y_{obs})}$ is more likely to occur if the corresponding $\boldsymbol{\theta}$ is positive.

The strength of ERGM is its capability of modeling endogenous interdependencies (*i.e.*, relations) among entities (*e.g.*, products) with various forms of network statistics, *i.e.*, $\boldsymbol{g(Y_{obs})}$, in addition to exogenous attributes pertaining to nodes and/or edges. Typically, the network statistics can be categorized into three main categories, *i.e.*, nodal attribute effects, relational attribute effects, and network structural effects (Morris et al., 2008), as shown in Table 2.3. Nodal attribute effects refer to the main effects of the nodes, which can be either continuous covariates or discrete (*e.g.*, categorical) variables. In a vehicle competition network, nodal attributes could be car features (*e.g.*, price, engine size). Relational attribute effects measure the effects of dyads' (*i.e.*, a group of two nodes) and edges' attributes. Examples of relational attributes include the similarity of dyad attributes and edge covariates. Moreover,

Table 2.3: Examples of three major categories of network statistics in ERGMs.

Categories	Examples	Interpreted effects
Nodal attributes effects		Car attributes (price, fuel consumption).
Relational attributes effects		Two cars with comparable attributes engage in competition against each other.
Network structural effects		Network density.
		Star effect.

* The hollow shape refers to the product w/o attributes. The solid shape refers to the product w/ attributes.

network structural effects measure the different levels of complexity of network structures, including the basic terms that control the overall probability of a link (such as the number of edges and density of a network), degree and star attributes which capture the distribution of node-based edge counts, and triangles and higher-order cycles that measure the effect of more complex local network structures.

2.3.3 Graph Neural Network (GNN)

Graphs are an important representation for complex systems (Rathkopf, 2018; Cui et al., 2020; Sha and Panchal, 2016) in that they not only model the interconnection and interrelation between system elements but also the leverage of complex network theories (Barabási, 2012). Graphs are non-Euclidean data, as opposed to other regular Euclidean data, such as images (2D grids) and texts (1D sequences). Its high dimensionality hinders the direct usage of some advanced neural network models such as CNN. To fill this gap, a Graph Neural Network (GNN) (Scarselli et al., 2008) was proposed in 2008, and due to its outstanding performance, GNN has been widely

used across domains since then (Song et al., 2022; Ferrero et al., 2022). For example, Ahmed *et al.* (Ahmed et al., 2021) developed a GNN-based method to predict the competition relationships between different car models in a vehicle co-consideration network. The model provided great insight into the key engineering attributes that promote the formation of car competitions.

The fundamental idea of GNN is that each node within a network is defined by its features and network neighbors, so each node in a network can be represented by these two pieces of information. Such a representation is also referred to as node embedding. Following the acquisition of node representation, various downstream tasks, such as node/link/graph classification, node/link/graph regression, node clustering, link prediction, and graph match, can be accomplished (Zhou et al., 2020). Recently, many variants of GNN have been developed, each based on a different node embedding strategy (Hamilton et al., 2017; Perozzi et al., 2014; Tang et al., 2015). For example, the well-known DeepWalk algorithm (Perozzi et al., 2014) generates node embedding in two steps, the first of which is to perform random walks on nodes in a graph to obtain node sequences. The skip-gram is then used in the second step to learn the node embeddings from the generated sequences (Chen et al., 2018).

GraphSAGE is another remarkable variant of GNN in that it is a general inductive framework. Unlike other frameworks that train individual embeddings for each node, GraphSAGE learns an embedding generating function by sampling and aggregating features from a node's neighborhood (Hamilton et al., 2017). This inductive framework provides a solution for graphs with varying node counts. Even if an unseen node is introduced into the graph, its representation can still be properly generated by feeding its neighborhood feature into the trained embedding generating function. A more detailed description of the algorithm can be found in (Hamilton et al., 2017).

Chapter 3: Network Motif-Based Robust Design of Socio-Technical System Against Seasonal Effects[1]

3.1 Overview

Given the nature of socio-technical systems (STS), which inherently involve numerous uncertain human behaviors, the imperative of designing reliable and robust STS has long been recognized. In this chapter, our objective is to develop a network motif-based robust design framework for STS to mitigate the impacts of seasonal effects. To achieve this goal, we introduce a novel approach grounded in network motif theory, aiming to optimize system capacity planning and resilience against fluctuations in demand due to seasonal changes. By leveraging the concept of motifs, we propose three key metrics for evaluating system performance and guiding capacity planning decisions: the imbalance score of a motif, a measure of motif seasonal robustness, and a design parameter-based criterion for capacity planning. We illustrate the application of our approach through a case study involving Divvy Bikes, a Chicago Bike Share program, demonstrating its effectiveness in enhancing system rebalance performance and robustness against seasonal variations. Finally, by realizing the objective of this chapter, ***RQ1***, ***RQ2***, and ***RQ3*** are answered.

The chapter is organized as follows:

- Section 3.2 introduces the proposed network motif-based robust design approach of STS against seasonal effects in detail.
- Section 3.3 presents the application of complex networks to model user behaviors in shared mobility systems (SMSs) and the process of network motif mining to identify the statistically significant travel patterns.

[1]The content of this chapter has been published in (Xiao and Sha, 2022). My contributions include conceptualization, methodology, formal analysis, and article writing.

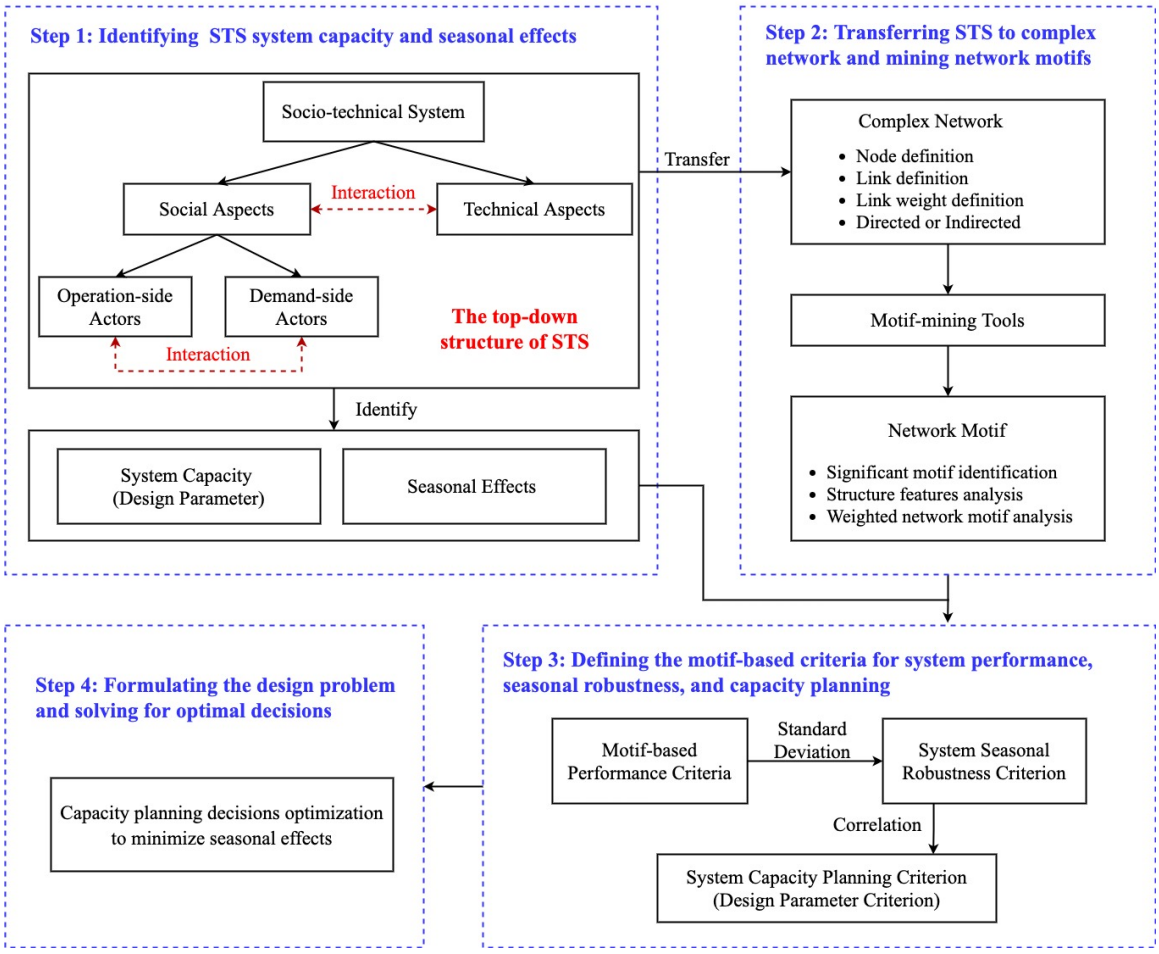


Figure 3.1: The framework for STS robust design against seasonal effects by capacity planning decisions optimization.

- Section 3.4 illustrates the assessment of the seasonal effect on the SMS's rebalance performance as well as formulates and solves the robust design problem in terms of SMS capacity planning.

- Section 3.5 provides the conclusions and major lessons learned from the study.

3.2 The Robust Design Approach

In this section, the proposed network motif-based robust design approach in a stepwise framework as shown in Figure 3.1 is introduced.

3.2.1 Step 1: Identifying STS System Capacity and Seasonal Effects

The main objective of this step is to formulate the seasonal robust design problem. It includes understanding the interconnections between different parts (as shown in the top-down structure) within an STS and identifying system capacity and seasonal effects. The in-depth understanding of the system helps lay down the foundation for the complex network construction in Step 2. During this step, data preprocessing is needed to organize the dataset by establishing the preprocessing tenets, *e.g.*, data preparation, cleaning, normalization and transformation of data, etc. (García et al., 2015).

3.2.2 Step 2: Translating STS to Complex Network and Mining Network Motifs

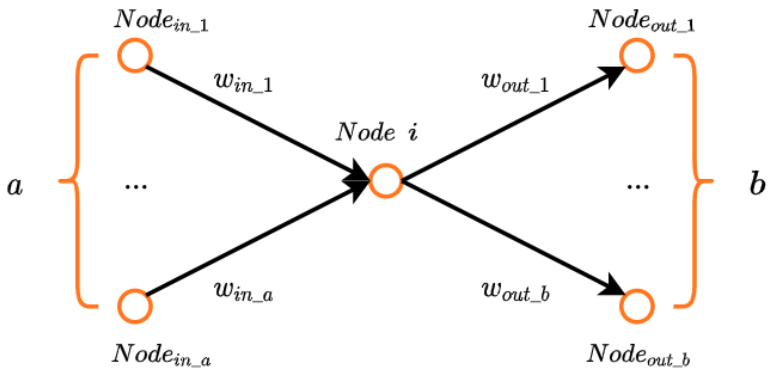
Based on the understanding of the target system and the robust design that needs to be addressed, the main tasks in Step 2 are to define and construct the complex network that best captures the STS structures as well as to mine the specific motif patterns in the established network. When building the complex network, we first need to determine the node, node features, link, whether the link carries weight or not, weight definition, and whether the network is directed or undirected. Secondly, since seasonal data are always time-dependent, two general strategies for handling such a temporal dynamic trait are often used. The first strategy is to treat the year-round data as time-series data, and the second one is to create cross-sectional data at different time steps. Since seasonal information typically changes by month, we adopt the second strategy with the information aggregated from each month. For example, one year's dataset can be divided into twelve cross-sectional datasets, which form twelve networks denoted as G_i $(i = 1, 2, ..., 12)$.

After the networks are constructed, motif mining tools like FANMOD (Rasche and Wernicke, 2006) and Mfinder (Kashtan et al., 2002) are employed to enumerate motifs with a particular size in each network. The significance scores (*i.e.*, *Z-score*

and *p-value*) of each pattern can be obtained at the same time. It is worth noting that during the motif mining, link weights are not used and motif patterns are mined only based on link existence. Link weights can be added later to the mined motif patterns for analysis if necessary. In our study, if a motif pattern is found significant in all the twelve networks, it is treated as a significant pattern throughout that entire year.

3.2.3 Step 3: Defining the Motif-based Criteria for System Performance, Seasonal Robustness, and Capacity Planning

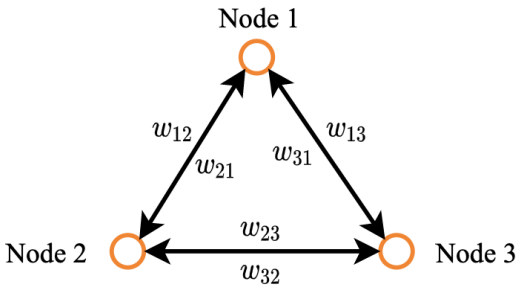
Before defining the criteria, we first introduce two node-level performance metrics for directed weighted networks. As shown in Figure 3.2, assuming a network G has T nodes and for any node $i \subset G$, there are a incoming weighted links and b outgoing weighted links. We define c as the difference between the sum of incoming link weights and the sum of outgoing link weights. Then, if $c = 0$, node i is defined as a *balanced node*; if $c < 0$, node i is considered to be *in-biased node*, and if $c > 0$, node i is named *out-biased node*. This way we are able to quantify a node's balance performance in STS.



$$c = \sum_{j=1}^{a} w_{in\_j} - \sum_{j=1}^{b} w_{out\_j} = \begin{cases} > 0 & \text{In-biased node} \\ = 0 & \text{Balanced node} \\ < 0 & \text{Out-biased node} \end{cases}$$

Figure 3.2: Categorizing a node based on its balance performance.

Next, we extend this node classification to network motifs. Supposed there is a size-3 motif comprising three nodes, *Node 1*, *Node 2*, and *Node 3*. A fully connected motif structure is shown in Figure 3.3. If weight $= 0$ can be used to represent a non-existent link (*e.g.*, if $w_{12} = 0$, it means there is no link from *Node 1* to *Node 2*), then all the thirteen size-3 directed motifs can be described with the following representation. According to the corresponding c values, *Node 1, 2, 3* are divided into three sets: $I(m)$, $O(n)$, and $B(l)$, where m, n, l represent the number of in-biased nodes, out-biased nodes, and balanced nodes in each set, and $m + n + l = 3$ holds. Moreover, the relationship among the three c values follows,



- For Node 1: $c_1 = (w_{21} + w_{31}) - (w_{12} + w_{13})$
- For Node 2: $c_2 = (w_{12} + w_{32}) - (w_{21} + w_{23})$
- For Node 3: $c_3 = (w_{13} + w_{23}) - (w_{31} + w_{32})$

Figure 3.3: A general motif structure.

$$c_1 + c_2 + c_3 = 0. \tag{3.1}$$

Based on these definitions, two metrics, α and β, are created (see Equations (3.2) and (3.3)) to grade every single motif in which α depicts the out-biased score and β indicates the in-biased score.

$$\alpha = \frac{1}{n} \sum |c_O|, \tag{3.2}$$

$$\beta = \frac{1}{m} \sum |c_I|, \tag{3.3}$$

where c_O or c_I indicates the nodes' c values that falling into the set $O(n)$ or $I(m)$. The higher the scores are, the less balanced is the motif. Based on Equation (3.1), it

can be proved that a linear relation exists between α and β (see Appendix A). The advantages of adopting these two metrics are reflected in two aspects. First, they are good indicators of the local-level system performance and can help designers locate the worst-performed sub-patterns. Second, not only can these two criteria capture the link weights, but they also integrate the topological characteristics of specific motifs.

At the system level, assuming a complex network G consists of K number of motif g (g is the motif ID in Table 2.2), two motif-based system performance criteria can be obtained below. Similarly, $\overline{\alpha}_g$ and $\overline{\beta}_g$ hold a linear relationship, and a higher value indicates a worse balance performance.

$$\textbf{\textit{Out-biased score}}: \quad \overline{\alpha}_g = \frac{1}{K}\sum_{j=1}^{K}\alpha_{g,j}, \tag{3.4}$$

$$\textbf{\textit{In-biased score}}: \quad \overline{\beta}_g = \frac{1}{K}\sum_{j=1}^{K}\beta_{g,j}. \tag{3.5}$$

Based on the motif-based system performance criteria, the ***seasonal robustness criterion***, as a quantitative representation of the seasonal effect, is defined as the standard deviation of the year-round in- or out-biased score of a motif [2]. For example, according to Step 2, we can get twelve monthly networks G_i ($i = 1, 2, ..., 12$) and the yearly significant motif patterns. For each significant motif, its aggregated in- or out-biased score over the twelve consecutive months can be calculated, and the resulting standard deviation from the twelve months, therefore, indicates the system robustness against seasonal changes.

Finally, we define the ***capacity planning criterion*** based on the capacity (v) of each service node in a network G. We denote the average capacity difference of a motif as

[2]Because of the linear relationship, the year-round distributions of $\overline{\alpha}_g$ and $\overline{\beta}_g$ should have a consistent trend, and only the amplitudes are different.

$$d = \frac{|v_1 - v_2| + |v_1 - v_3| + |v_2 - v_3|}{3}, \tag{3.6}$$

where v_i $(i = 1, 2, 3)$ is the capacity of each node i in a size-3 motif. Correspondingly, in the network G consisting of K motif g, the average capacity difference of motif g is

$$\overline{d}_g = \frac{1}{K} \sum_{j=1}^{K} d_{g,j}. \tag{3.7}$$

To provide more insights into how the three motif-based metrics be utilized and extended to different systems, a quick overview of the metric interpretations along with their application examples are summarized in Table 3.1

3.2.4 Step 4: Formulating the Design Problem and Solving for Optimal Decisions

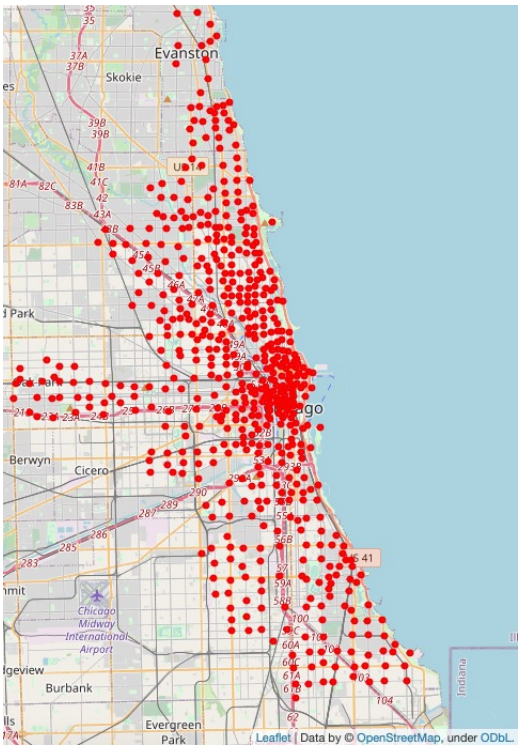
This step's objectives are two-fold: 1) investigate the correlation between seasonal effects (represented by the seasonal robustness criterion) and the capacity planning criterion, as identified in Step 3, and 2) formulate the design problem and solve it to obtain optimal decisions for improving the system's robustness against seasonal disturbance. It would be ideal that the factors influencing the system's robustness are known from existing domain knowledge, so such factors will be formulated into the design problem as the decision variable to be optimized. Otherwise, correlation analysis and/or causal inference need to be applied to identify the key design variables.

3.3 Shared Mobility System Trip Network Modeling and Network Motif Mining

In this study, the Chicago Bike Share program, Divvy Bikes, is selected to demonstrate the proposed approach. The Divvy Bikes' data is publicly achievable (Divvy_Bike, 2020), and the data from 2014 to 2017 are adopted due to the availability of capacity

Table 3.1: The interpretations of the motif-based metrics in different applications.

Metrics	Interpretation	Application Examples
Imbalance Score	The ***imbalance score*** is proposed to quantitively describe the local service networks' rebalance performance, *i.e.*, the difference between the inflow and out-flow information/traffic of a local system. This metric can quantitively evaluate the local service networks' performance of STSs.	• In the interconnected power grid, the ***imbalance score*** is the average difference of transmitting-in and -out power within a local service power grid. • In the SMS, the ***imbalance score*** is the average difference of numbers of rental and return bikes of a local-level service system (*e.g.*, a system including three service stations).
Seasonal Robustness Criterion	The ***seasonal robustness criterion*** is the standard deviation of the year-round imbalance score. It is a quantitative representation of the seasonal effect where a larger value indicates a local service system is more sensitive to seasonal demand fluctuation.	• In the interconnected power grid, the ***seasonal robustness criterion*** represents how the average difference of transmitting-in and -out power of a local service power grid fluctuates along with seasonal changes. • In SMS, the ***seasonal robustness criterion*** represents the variation of the average difference of the rental and return bikes within a local service system along with seasonal changes.
Capacity Planning Criterion	The ***capacity planning criterion*** describes the average capacity difference in a local service network. It is an efficient indicator of whether the local service system's resource distribution is balanced or not.	• In the interconnected power grid, the ***capacity planning criterion*** is the average difference of the maximum energy storage ability within a local power grid. • In SMS, the ***capacity planning criterion*** is the average difference of the dock numbers within a local service system.



System Name: Divvy Bike

Locale: Chicago

Date of Operation Began: June 28, 2013

Year	Number of Station
2014	300
2015	475
2016	581
2017	585

Figure 3.4: Divvy Bike system information.

information (*i.e.*, the number of docks) at each station. Figure 3.4 shows the station distribution of Divvy Bikes in the third and fourth quarters of 2017 and the number of stations in each year. In this study, we aim to mitigate the sensitivity of the system's rebalance performance to seasonal effects.

3.3.1 Data Preprocessing

The station and trip data packages contain information like station geographic coordinates, the number of docks, trip start and end station IDs, trip time and duration, and user basic information (*e.g.*, gender and birth year). We follow four steps to process the raw data for each year. The final data frame consists of twelve monthly trip datasets, each of which has three columns, including start station ID, end station ID, and the reoccurring frequency of each unique trip.

1) Basic trip information extraction. The essential data, such as the trip start and end station IDs, the number of docks, and start and end times are extracted from the raw dataset.

2) Data cleaning. We delete those trips with missing data and the testing stations (*e.g.*, station 512 is a station for a testing purpose only) along with their associated trips.

3) Monthly trip network data preparation. In this step, we split trips by month based on their starting time.

4) Trip reoccurring frequency calculation. We count the number of times that a trip between a pair of stations occurs in each month.

3.3.2 Trip Network Building and Motif Mining

Based on the monthly trip datasets, the monthly trip networks are constructed. In each network, stations are represented as nodes, and a trip between two stations is defined as a link and its reoccurring frequency in a month is the link weight. Since the trip from Station A to Station B is different from the trip from B to A, the resulting trip network is a directed network.

To focus on the network that captures the most significant traffic, we delete those links that have less occurred trips, such as those links with just one-time transit. The threshold for such a link removal process is set as the minimum mean (u) of the link weights among the twelve-monthly trip networks in the interested years:

$$u = Min(u_{ij}),\ i = \text{index of years}, j = 1, 2, ..., 12, \tag{3.8}$$

where u_{ij} is the link weight mean of the j^{th} network in year i. For example, from 2014 to 2017, $u = 3.03$. Then, all the links with weights lower than 3.03 are removed from the network. Figure 3.5 illustrates the link weight distribution of Divvy Bikes

in July 2017. It reveals that the statistical features of link weights will not be altered by removing the links with weights below the threshold. Figure 3.6 shows the visualization of the reduced trip network.

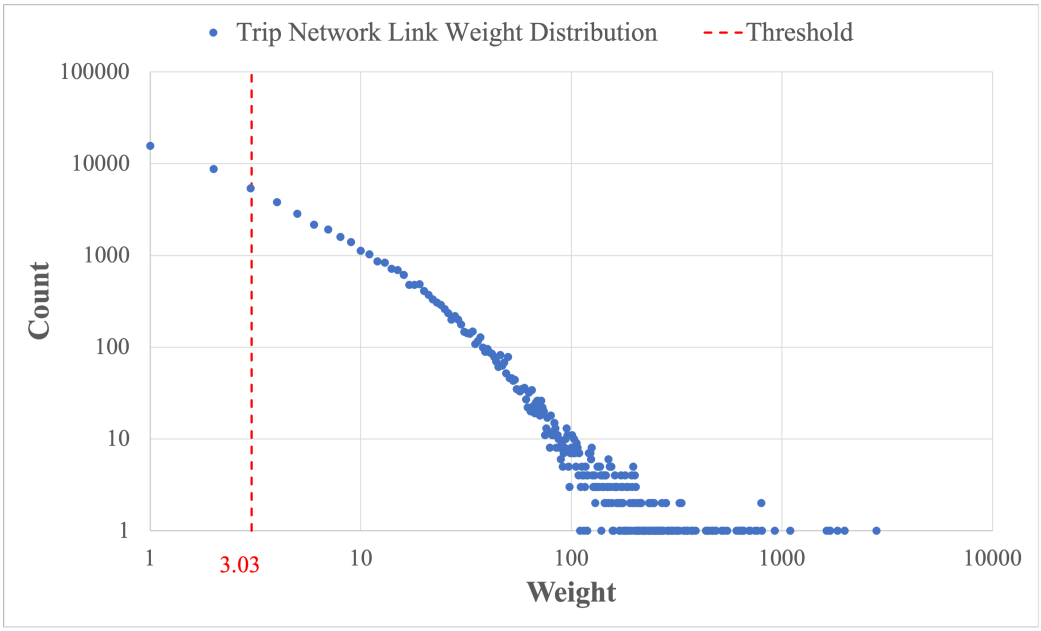
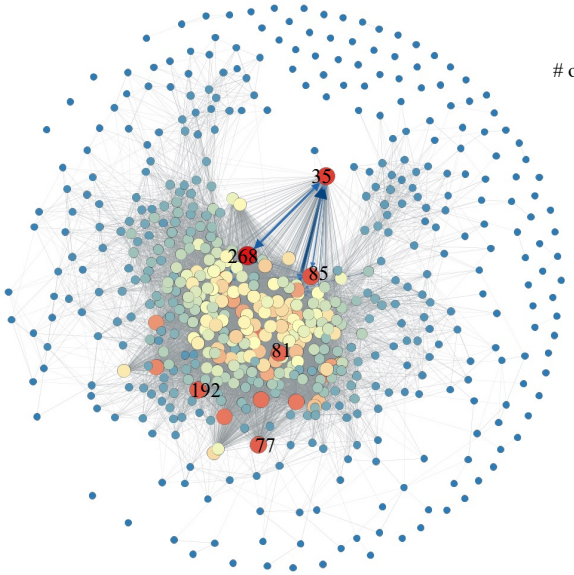


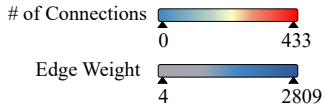
Figure 3.5: Weight distribution of Divvy Bike trip network (Jul 2017, total edges: 57225).

After obtaining the weighted directed trip networks, their binary counterparts (*i.e.*, the same network without link weights) are used for motif mining, which reports the motif structures, Z-scores, p-values, and the adjacent matrix list of all existing motifs. In this study, the motif mining tool FANMOD is adopted. Table 3.2 shows the thirteen size-3 directed motif IDs in every month of 2017, ranked from top to bottom based on their Z-scores from high to low. The red IDs are insignificant motifs under the level of significance .001.

According to the results, we observe that the motifs with high transitivity[3] are

[3]A triad involving nodes i, j, and k is transitive if whenever i connects to j and j connects to k then i connects to k (Wasserman and Faust, 1994). A digraph has high transitivity if most triads it contains are transitive.





Top Six Hub Stations

Station ID	Station Name	# of Connections
268	Lake Shore Dr & North Blvd	433
35	Streeter Dr & Grand Ave	398
85	Michigan Ave & Oak St	376
192	Canal St & Adams St	368
77	Clinton St & Madison St	367
81	Daley Center Plaza	359

Figure 3.6: A visualization of Divvy Bike trip network after removing the links with fewer occurred trips (Jul 2017, total edges: 27415).

more likely to be significant and ranked higher in the trip network. This is also the reason that motif 78 is always ranked lowest in all networks. A similar phenomenon is also observed in the years from 2014 to 2016, as shown in Appendix B. In the following analysis, only the significant motif patterns over two years, including motifs 238, 102, 174, 166, 38, 46, and 140, are considered.

3.4 Shared Mobility System Robust Design to Against Seasonal Effects

3.4.1 Identifying SMS Design Parameters and Seasonal Effect

In our prior work (Xiao and Sha, 2020), it is found that seasonal changes can influence the average distances of trip motifs. For example, users tend to ride longer distances in warmer seasons. Moreover, seasonal changes can impact the traffic of local networks, which is a critical factor in the system rebalance performance. As to the design parameter, based on the correlation analysis (see Table 3.4), it is found that the number of docks at each station plays an important role in the rebalancing problem

Table 3.2: Divvy Bike motif Z-score ranks of each month in 2017.

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
238	238	238	46	46	166	46	46	46	46	238	238
102	102	102	102	102	102	166	166	166	102	102	102
46	46	46	166	166	238	102	102	102	166	166	166
174	166	166	238	238	38	238	238	238	238	46	46
166	174	174	38	38	140	38	38	38	38	174	174
38	38	38	174	174	46	140	140	140	174	38	38
140	140	140	140	140	12	12	174	174	140	140	140
12	12	12	12	12	174	174	12	12	12	12	12
6	6	6	6	14	14	14	14	6	6	6	36
36	36	36	14	6	164	164	6	14	14	36	6
164	14	14	164	164	6	6	164	164	164	14	164
14	164	164	36	36	36	36	36	36	36	164	14
78	78	78	78	78	78	78	78	78	78	78	78

because it directly relates to the availability of bikes that a user can rent or return in a station.

Figure 3.7 shows the average dock difference of those significant motifs in the twelve months of 2017 following Equations (3.6) and (3.7). It is observed that the average dock difference curves from top to bottom correspond to the rank of transitivity of the motifs from high to low. For example, motif 238, the pattern with the highest transitivity in the trip networks, has the largest dock difference in the entire year of 2017, while motifs 140 and 38 have the smallest differences. While the causation needs to be further investigated, one possible reason for such a correlation could be that the stations with large capacities are more likely located in high-demand areas, thus more users will return or rent bikes. So, they are hub stations and will connect to many other stations. Meanwhile, there is majority of stations within the system have low capacities. Therefore, these stations and hubs form a large proportion of motifs with high transitivity and large capacity differences.

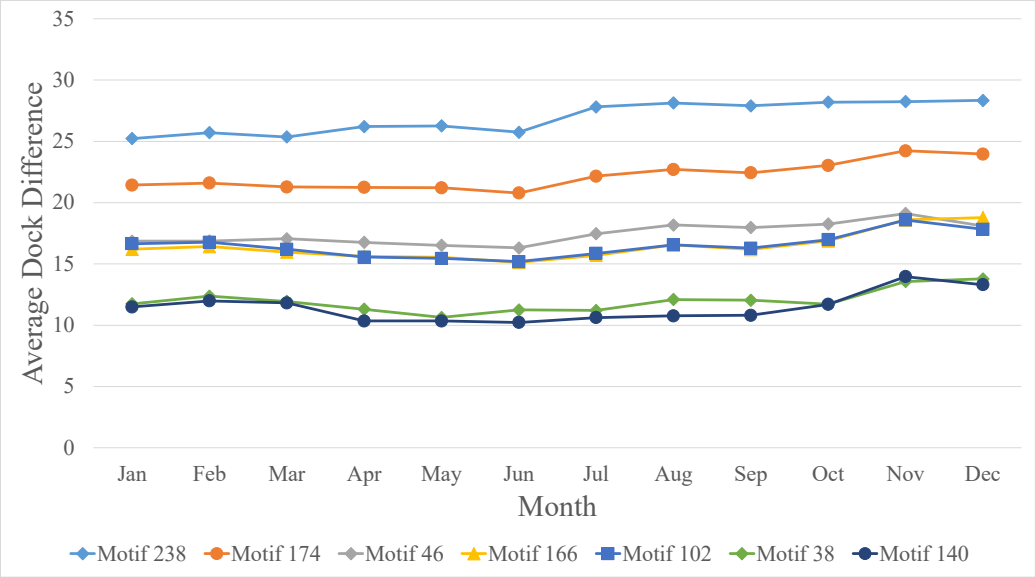
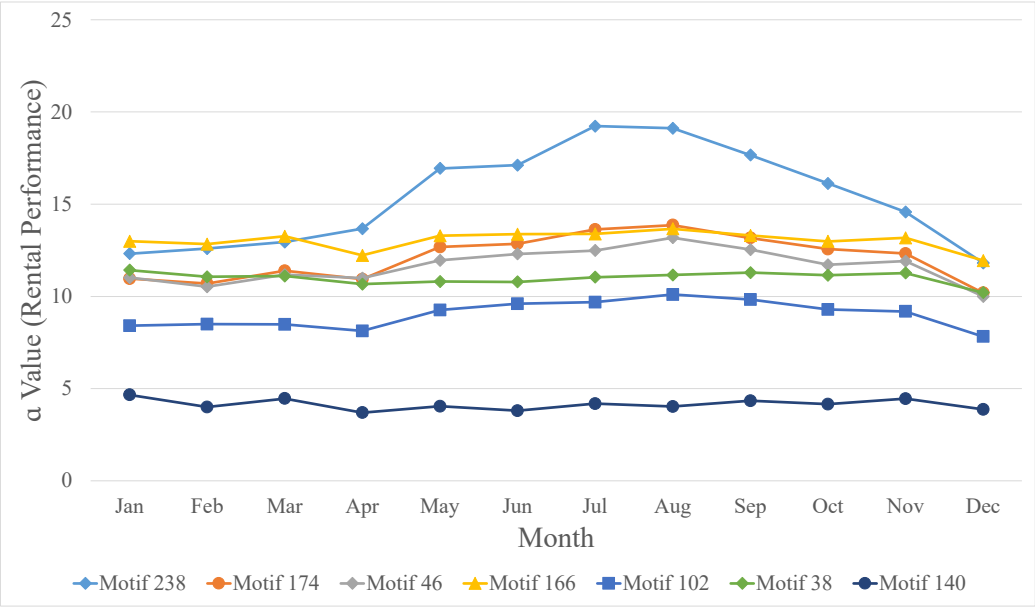


Figure 3.7: Divvy Bike yearly motif dock difference curves (2017).

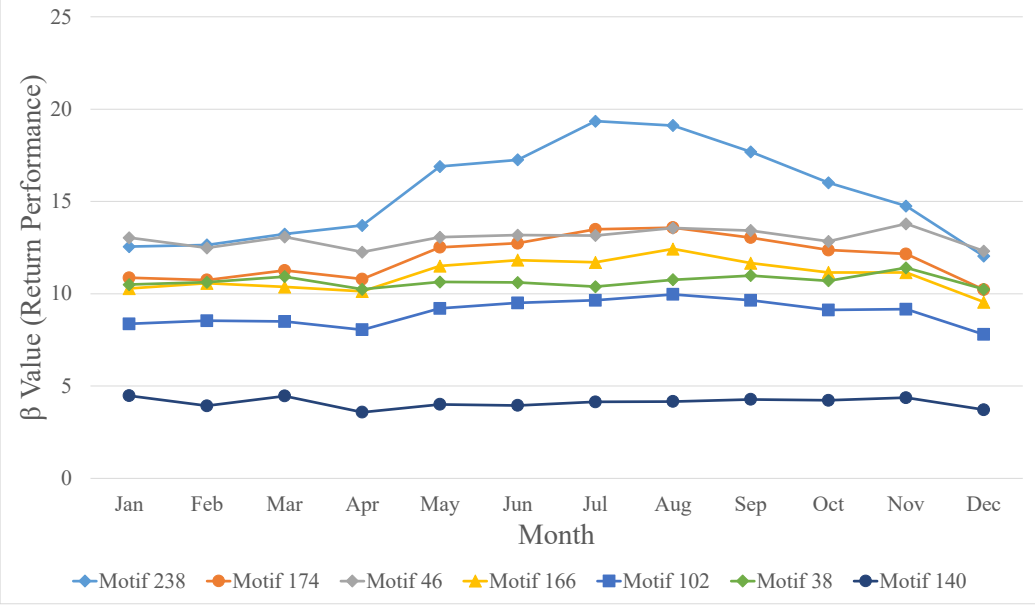
3.4.2 Trip Motifs Performance and Robustness Analysis

In this section, by following Step 3 in Section 3, we calculate the local SMS rental and return performance scores, which correspond to the motif-based in- and out-biased values. In an SMS, a higher rental or return score indicates that a serious rebalance issue could occur in a trip motif. Figure 3.8 shows the rebalance performance scores of the seven significant trip motifs.

As indicated in both Figure 3.8 (a) and (b), the trip motif's rebalance performance is potentially related to the motif structure. Taking motifs 46, 166, and 140 as examples, both motifs 46 and 166 have apparent unbalanced structures where *Node 1* only has in-arrows or out-arrows (see Figure 3.9). This leads them to be vulnerable to return and rental problems. In contrast, the number of in- and out-arrows of all nodes in motif 140 are the same, so motif 140 is expected to have a low rebalance performance score. However, there are also a few exceptions. For example, motif 238 has a balanced structure but still experiences return and rental problems for several months from April to November. These abnormal fluctuations remind us of the potential seasonal effects, so we use the standard deviation of the return/rental



(a) Rental performance



(b) Return performance

Figure 3.8: Divvy Bike yearly motif rebalance performance (2017).

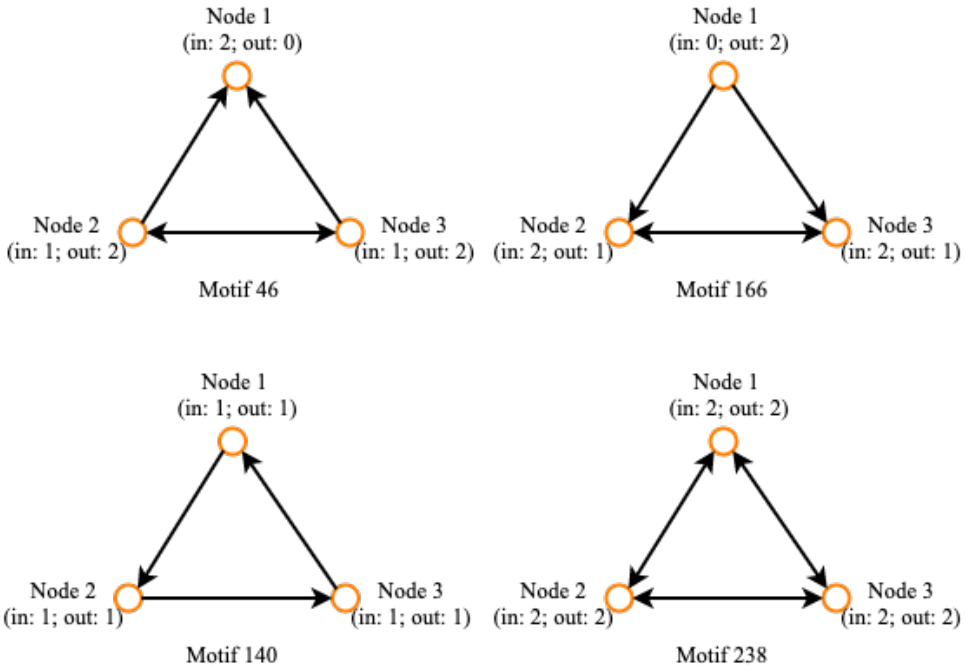


Figure 3.9: Trip motif structure analysis.

Table 3.3: Divvy Bike seasonal robustness criteria and capacity planning criteria of significant trip motifs (2017).

Motif ID	238	174	46	166	102	38	140
Seasonal robustness score (the standard deviation of β value)	2.622	1.160	0.476	0.847	0.694	0.331	0.282
Capacity planning criterion	26.931	22.177	17.458	16.467	16.497	11.974	11.452

performance scores in a year to quantify such fluctuations, as shown in the second row of Table 3.3. A larger deviation means that a trip motif is more sensitive to seasonal changes.

3.4.3 Design Problem Formulation

To confirm the targeted design variable, we first conducted a correlation analysis between the system robustness and the average capacity difference. Since the robustness score is measured based on yearly data, the mean of the capacity differences of every

significant motif during the entire year is calculated, as listed in the third row of Table 3.3. Based on this table, the correlation coefficient between the system robustness and the capacity planning criterion in 2017 can be obtained, and similarly, for the data from 2014 to 2016.

The results are summarized in Table 3.4 and reveal a significantly high correlation between the capacity difference and the system's robustness. In other words, if a trip motif has a large average capacity difference, its rebalance performance would be more sensitive to seasonal changes. This observation has led to our design objective - to optimize the capacity of the stations in the motifs that are most influential to the system's robustness. To this end, we split the task into two sub-tasks: 1) identify the stations that need to be optimized for their number of docks, and 2) plan the capacity, *i.e.*, the number of docks for those stations, either by adding docks or removing docks, to minimize the average dock difference.

Table 3.4: Divvy Bike yearly correlation coefficient between seasonal effect and motif dock differences.

Year	2014	2015	2016	2017
Correlation coefficient	0.848	0.921	0.914	0.922

In the first sub-task, the motif pattern that is the most sensitive to seasonal changes is chosen (assuming its ID is g_{season}). Then, we determine the objective motifs with the largest dock differences every month and identify the station IDs that construct those motifs. Based on the number of times those identified stations appear in the objective motifs in each month, two decision rules[4] are used to decide which stations' capacity needs to be optimized.

- The first rule is that among all twelve months, if a station appears in the most number of months, then it will be regarded as the critical station and its capacity

[4]These two rules can be extended. For example, instead of choosing the most frequently appearing stations, the most and the second most frequently appearing stations can be chosen in both rules to achieve more deduction in capacity difference.

will be taken into account for optimization. From the first rule, we will identify a set of critical stations, S_1.

- In the second rule, the stations appearing most frequently in each month are chosen as critical stations, and the corresponding station set is defined as S_2. Finally, we define all the critical stations being represented as $S = S_1 \bigcup S_2$.

In the second sub-task, assuming the significant motif set is M, including m different types of motifs, we identify the significant motifs (from M) in which a critical station s ($s \in S$) appears, and put the same type of motifs with the ID $g \in M$ in one set, $M_{s,g}$. Next, we define the decision variable x_s as the number of docks that station s needs to add ($x_s > 0$) or remove ($x_s < 0$). Then, the updated average capacity difference of the motif g, $d_{s,g,j}$, can be calculated by following Equation (3.9), where s_1, s_2, and s_3 represent three stations' IDs in a motif. Depending on whether s_1, s_2, and s_3 belong to the critical station set S or not, $d_{s,g,j}$ is calculated differently.

$$\begin{cases} d_{s,g,j}(x_{s_1}) = \frac{1}{3}[|(v_{s_1} + x_{s_1}) - v_2| \\ \qquad\qquad + |(v_{s_1} + x_{s_1}) - v_3| + |v_2 - v_3|] & \text{if } s_1 \in S \\ d_{s,g,j}(x_{s_1}, x_{s_2}) = \frac{1}{3}[|(v_{s_1} + x_{s_1}) - (v_{s_2} + x_{s_2})| \\ \qquad\qquad + |(v_{s_1} + x_{s_1}) - v_3| + |(v_{s_2} + x_{s_2}) - v_3|] & \text{if } s_1, s_2 \in S \\ d_{s,g,j}(x_{s_1}, x_{s_2}, x_{s_3}) = \frac{1}{3}[|(v_{s_1} + x_{s_1}) - (v_{s_2} + x_{s_2})| \\ \qquad\qquad + |(v_{s_1} + x_{s_1}) - (v_{s_3} + x_{s_3})| & \text{if } s_1, s_2, s_3 \in S \\ \qquad\qquad + |(v_{s_2} + x_{s_2}) - (v_{s_3} + x_{s_3})|] \end{cases} \tag{3.9}$$

Finally, the updated average dock difference for motifs in set $M_{s,g}$ can be obtained

$$\overline{d}_{s,g} = \frac{1}{m_{s,g}} \sum_{j=1}^{m_{s,g}} d_{s,g,j}, \tag{3.10}$$

where $m_{s,g}$ is the number of motifs in $M_{s,g}$.

Since the objective is to minimize the average dock difference of those identified trip motifs, a multi-objective optimization is formulated in Equation (3.11):

$$\begin{cases} \min \overline{d}_{s_1,g_1} = \min \frac{1}{m_{s_1,g_1}} \sum_{j=1}^{m_{s_1,g_1}} d_{s_1,g_1,j} \\ ... \\ \min \overline{d}_{s_1,g_m} = \min \frac{1}{m_{s_1,g_m}} \sum_{j=1}^{m_{s_1,g_m}} d_{s_1,g_m,j} \\ ... \\ \min \overline{d}_{s_l,g_1} = \min \frac{1}{m_{s_l,g_1}} \sum_{j=1}^{m_{s_l,g_1}} d_{s_l,g_1,j} \\ ... \\ \min \overline{d}_{s_l,g_m} = \min \frac{1}{m_{s_l,g_m}} \sum_{j=1}^{m_{s_l,g_m}} d_{s_l,g_m,j} \end{cases} \tag{3.11}$$

$$\text{S.T. } x_{s_1} \geq -v_{s_1}, ..., x_{s_l} \geq -v_{s_l} \text{ and } x_{s_1}, ..., x_{s_l} \in \boldsymbol{Z},$$

where $s_1, ..., s_l \in S$, $g_1, ..., g_m \in M$, $v_{s_1}, ..., v_{s_l}$ are the original dock numbers of station $s_1, ..., s_l$. $\boldsymbol{Z}$ denotes Integer. In Equation (3.11), all the relevant motifs in M, even if they are not g_{season}, are considered. This is because while we are changing the number of docks for those stations in motif g_{season}, there is a possibility that the average dock difference in the other motifs which include the stations of g_{season} increases too.

To solve this optimization problem, we adopt the weighting method (Miettinen, 2012) to transform the multi-objective optimization problem to a single-objective one in Equation (3.12). Suppose all objective functions in Equation (3.11) are equally important, and $\sum_{i=1}^{m \times l} q_i = 1$, then $q_i = q = \frac{1}{m \times l}$ $(i = 1, ..., m \times l)$. Equation (3.12) is a typical nonlinear integer optimization problem, and the genetic algorithm, *ga* function in (MATLAB, 2020) is applied to solve this problem.

$$\begin{gathered} \min D(x_{s_1},...,x_{s_l}) = \min[q(\overline{d}_{s_1,g_1} + \overline{d}_{s_1,g_m} + ... + \overline{d}_{s_l,g_1} + \overline{d}_{s_l,g_m})], \\ \text{S.T. } x_{s_1} \geq -v_{s_1}, ..., x_{s_l} \geq -v_{s_l} \text{ and } x_{s_1}, ..., x_{s_l} \in \boldsymbol{Z}, \end{gathered} \tag{3.12}$$

Table 3.5: The calculating results of Equation (3.12).

Station ID	3	35	45	97	172	263
Original dock number	55	47	15	55	11	11
Added/deleted dock number	-11	-11	8	-32	12	12
Updated dock number	44	36	23	23	23	23

*: correspond to $x_{s_1}, ..., x_{s_6}$

In our case study, based on Figure 3.8, we identify $M = \{238, 174, 46, 166, 102, 38, 140\}$ and motif 238 is the target motif we need to focus on because it is the most sensitive one in light of seasonal changes. Table 3.7 lists most of the critical stations that form motif 238 and yield the largest dock difference. From Table 3.7, we can observe that, station 3, the most frequently appeared station (nine months out of 12), should be considered as a critical station, *i.e.*, $S_1 = \{3\}$. Regarding the stations that appear most in each month, taking March as an example, we identified 15 critical motif 238s, and stations 35 and 172 are the most frequently appeared stations in all of the 15 critical motifs. Thus, they are considered as critical stations. Similarly, another four critical stations are identified, thus $S_2 = \{3, 35, 45, 97, 172, 263\}$. By combining these two sets, we obtain the final critical station set $S = S_1 \bigcup S_2 = \{3, 35, 45, 97, 172, 263\}$.

By solving the optimization problem in Equation (3.12), we obtain the optimal capacity planning decision for the decision variables x_{s_1},...,x_{s_6}. The results are shown in Table 3.5, along with the original and updated number of docks. To verify if the redesigned capacity can effectively decrease the average dock difference of the significant trip motifs or not, we recalculate the trip motifs' mean values of the updated number of docks in a year, as shown in Table 3.6. By comparing the updated dock differences with the original ones, it is found that the decreases are achieved for all significant motifs, and the dock difference of motif 238 is decreased by 4.6%. With such a decrease, the enhancement of the system robustness against seasonal effects is expected to be achieved effectively.

Table 3.6: Divvy Bike yearly mean values of significant motif dock differences, before update vs after update (2017).

Motif ID	238	174	46	166	102	38	140
The mean of the motif dock difference before update	26.931	22.177	17.458	16.467	16.497	11.974	11.452
The mean of the motif dock difference after update	25.679	21.199	16.758	15.810	15.832	11.557	11.044
The percentage of decrease (%)	4.6	4.4	4.0	4.0	4.0	3.5	3.6

3.5 Conclusion

It is the uncertainty and complicated interactions within an STS that make the system vulnerable to various perturbations. The occurrence of certain perturbations can significantly influence STS performance, and the seasonal effect is a common one because it directly impacts human behavior in STS. In this chapter, we develop a new design framework for improving STS robustness against seasonal changes based on the network motif theory. Using the concepts of motif, we created three metrics for system performance evaluation and capacity planning decision-making. The first one is the imbalance score of a motif (*e.g.*, a local service network), the second one is the measurement of a motif's seasonal robustness, and the third one is a design parameter-based capacity planning decision criterion. We apply our developed approach to a real-world STS, Divvy Bikes, a Chicago Bike Share program, to improve the system's rebalance performance and its robustness against seasonal changes. The results from this study show that our approach can effectively reduce the average dock differences among the stations of critical trip motifs (*i.e.*, local trip networks), thereby improving the system's robustness.

The main contributions of this chapter are summarized in three aspects: 1) We introduce a network motif-based approach for guiding the STS robust design, emphasizing optimizing system capacity planning to weaken the impact of demand fluctuations caused by seasonal changes 2) We propose a set of motif-based criteria to

Table 3.7: Station list of constructing the motif 238s with the largest dock difference values*.

Jan		Feb		Mar		Apr		May		Jun		Jul		Aug		Sep		Oct		Nov		Dec	
Station ID	Frequency	Station ID	Frequency	Station ID	Frequency	Station ID	Frequency	Station ID	Frequency	Station ID	Frequency	Station ID	Frequency	Station ID	Frequency	Station ID	Frequency	Station ID	Frequency	Station ID	Frequency	Station ID	Frequency
35	5	35	20	35	15	97	82	97	138	97	255	3	25	3	10	3	5	26	26	3	26	3	6
172	5	172	19	172	15	35	75	172	95	172	149	42	17	263	10	263	5	172	26	24	18	45	6
59	2	97	11	25	1	172	47	35	89	35	134	263	8	406	1	406	1	268	1	45	11	338	3
97	2	45	7	321	1	288	23	195	43	192	88	623	2	338	1	150	1	264	1	97	8	97	3
25	1	150	4	301	1	24	22	24	29	24	45	338	2	287	1	76	1	211	1	59	3	59	2
45	1	76	2	286	1	199	21	199	29	199	44	237	2	255	1	62	1	199	1	90	3	37	1
194	1	255	2	283	1	45	17	45	28	45	39	150	2	237	1	4	1	196	1	255	2	90	1
176	1	4	1	211	1	338	17	288	25	288	37	62	2	150	1			194	1	195	2	255	1
338	1	6	1	199	1	150	11	173	21	338	36	370	1	76	1			192	1	174	2	192	1
126	1	24	1	176	1	192	11	192	20	173	35	178	1	62	1			181	1	91	2	168	1
99	1	25	1	173	1	146	8	338	19	112	29	147	1	6	1			177	1	52	2	91	1
		26	1	161	1	76	6	150	15	370	23	120	1	4	1			173	1	43	2	43	1
		291	1	145	1	90	6	146	11	150	23	91	1					161	1	35	2		
		289	1	141	1	3	5	370	10	146	20	81	1					142	1	26	2		
		286	1	99	1	43	5	76	9	3	10	72	1					99	1	370	2		
		268	1	77	1	49	5	90	8	76	10	59	1					91	1	338	2		
		211	1	47	1	85	5	6	7	85	10	45	1					90	1	341	1		
		199	1			284	5	26	7	90	10	255	1					85	1	284	1		
		181	1			255	5	43	7	2	9	76	1					76	1	264	1		
		177	1					52	7	6	9	4	1					52	1	194	1		
		173	1					268	7	43	9	41	1					49	1	47	1		
		168	1					2	6	52	9	33	1					38	1	44	1		
		99	1					3	6	91	9	5	1					37	1	41	1		
		341	1					85	6	255	9							36	1	25	1		
		164	1					99	6	26	8							35	1	623	1		
		145	1					284	6	36	8							26	1	4	1		
		141	1					255	6	100	8							25	1				
		126	1					4	5	268	8							24	1				
		110	1					62	5	264	8												
		94	1					142	5	181	8												
		47	1					145	5	4	7												
		90	1					341	5	37	7												
								309	5	44	7												
										99	7												
										419	7												
										211	7												
										196	7												
										195	7												
										321	7												
										33	6												
										59	6												
										62	6												
										75	6												
										145	6												
										341	6												
										284	6												
										177	6												
										623	5												

: Among twelve months, stations that appear in the most number of months

: Stations that appear most in each month

*: Due to the space limitation, stations with appearing frequencies less than 5 in April, May, and June are ignored, and this ignorance has no effect on critical station identification.

help evaluate system's performance and the impact of seasonal effects on it. 3) A high correlation between the seasonal effects and the average dock difference of motifs is discovered in BSS, from which a multi-objective design problem is formulated to aid capacity planning decisions for improved system robustness. The process of significant trip motif mining contributes to addressing ***RQ1***. Furthermore, the development of three trip motif-based metrics for evaluating system performance and making capacity planning decisions contributes to answering ***RQ2*** by exploring the influence of meso-level subsystems on macro-level system performance. Lastly, the formulation and resolution of the seasonal robust optimization design problem contribute to addressing ***RQ3***.

Chapter 4: Graph Neural Network-Based Design Decision Support for Socio-Technical Systems[1]

4.1 Overview

This chapter aims to develop a complex network-based approach rooted in Graph Neural Network (GNN) techniques for predicting links within Socio-Technical Systems (STSs). Since GNN's core concept involves incorporating both entity attributes and local network neighborhood information to enhance prediction performance, the proposed approach contributes to addressing ***RQ2***. Furthermore, leveraging the predictive model derived from this approach, the chapter explores its utility as an effective tool for testing design strategies and aiding system designers in decision-making processes. This exploration contributes to addressing ***RQ3***. Finally, the chapter employs SMS for validation purposes.

The chapter is organized as follows:

- Section 4.2 introduces the proposed complex network-based prediction approach and the associated methods for model analysis and evaluation.
- Section 4.3 takes Divvy Bike in Chicago as an example to demonstrate the utility of the proposed prediction approach.
- Section 4.4 illustrates how the proposed predictive model can be utilized to support system design decisions.
- Section 4.5 discusses the limitations of the current study that can lead to future investigations.
- Section 4.6 concludes the study with closing thoughts.

[1]The content of this chapter has been published in (Xiao et al., 2023a). My contributions include conceptualization, methodology, formal analysis, and article writing.

4.2 Complex Network-Based Prediction Framework

An overview of the complex network-based approach to predicting travel demand for shared mobility systems is shown in Figure 4.1. In this approach, we start by modeling a shared mobility system as a complex network using historical data, *i.e.*, Period One data, including station attributes and trip data. After obtaining the network model, we utilize Period One data to train predictive models, including the ANN model in Section 4.2.2 and the GraphSAGE-based model in Section 4.2.3. The trained model is then employed to predict the network links in Period Two based on the updated nodal attributes. To evaluate the predictive performance of the models, the predicted links are compared with the actual ones, and the metrics quantifying the prediction accuracy are introduced in Section 4.2.4.

4.2.1 Node Attributes

In this study, the node represents the bike sharing station and the node attributes indicate the station features. The node attributes considered here include the geographic coordinates of the station, the number of docks, and the POIs surrounding the stations. Geographic coordinates can be used to calculate the distance between two stations, and the number of docks at a station determines the maximum number of bikes that users can rent from or return to that station. Current research indicates that there are evident travel patterns between certain functional zones of the city due to user-specific travel purposes (He and Shin, 2020; Liu et al., 2017). In the study (He and Shin, 2020), for example, He and Shin divided POI into five major categories, including residential, cultural, recreational, commercial, and governmental. They found that travel behavior in BSS has a stronger correlation between stations in recreational and residential areas than between stations in recreational and commercial areas.

In this study, POI data are collected by Overpass turbo (Wiki, 2022) which includes the name of each POI and its geographic coordinates. We first classify POIs into seven categories, including Financial, Education, Recreational&Tourism, Resi-

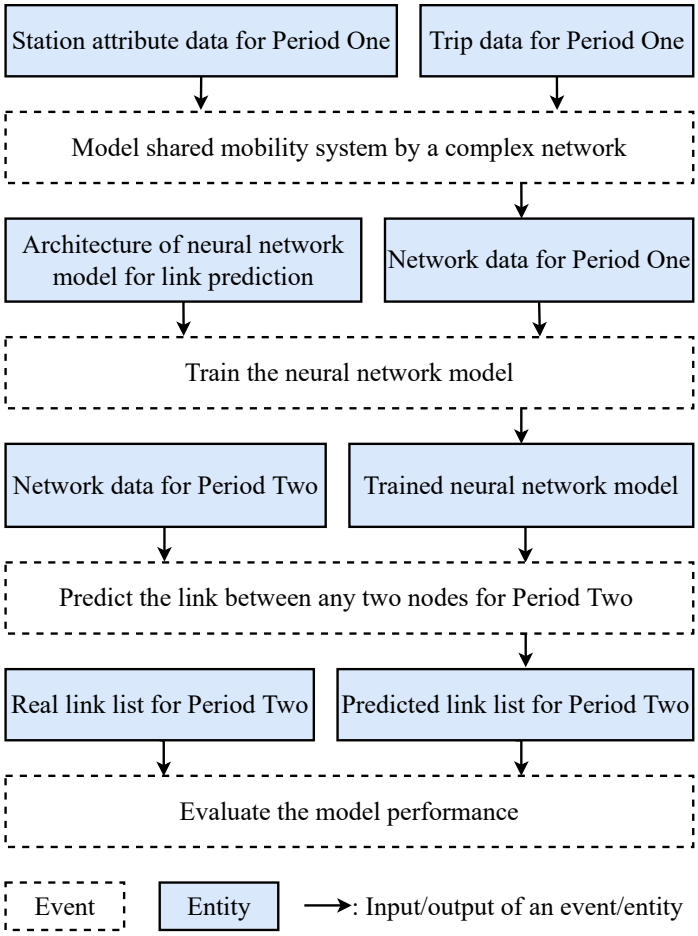


Figure 4.1: Complex network-based prediction framework for shared mobility systems design support with Neural Network (**Period One**: Month i in year Y, **Period Two**: Month i in year $Y+1$, $i = 1, ...12$)

dential, Sustenance, Healthcare, and Transportation. The details of these categories are given in (Yinshuang et al., 2022). Then, we draw a circle of radius R with the target station in the center of the circle. Finally, we count the number of POIs in each category within the circle and treat the combination of seven counts as an attribute vector of the target station. Regarding the value of the radius, R, in reference (Yang and Diez-Roux, 2012), the authors calculated the cumulative percentage distribution of walking trips by distance based on data from the 2009 U.S. National Household Travel Survey. We learned from the distribution that 1.5 miles is the walking distance upper bound of 90% of walking trips. Therefore, POIs within 1.5 miles of a station

provide the best representation of the station's surroundings. Taking Divvy Bike station 368 (Ashland Ave & Archer Ave) in Chicago as an example, its POI attribute vector in 2016 is $[2, 30, 0, 2, 4, 12, 12]$, indicating that there are two banks, thirty education institutions, two healthcare institutions, four apartments, twelve restaurants, and twelve public transportation stops within a radius of 1.5 miles.

4.2.2 Baseline: ANN-Based Link Prediction Model

In this study, we take ANN as the baseline model. As shown in Figure 4.2, the architecture of a simple ANN model consists of an input layer, one hidden layer, and an output layer. Training a model starts by formulating link features. In a shared mobility network, the link features are determined by the two connecting nodes. Accordingly, we use the concatenation of the features of the start and end nodes with size N to represent the features of the directed link with size $2N$. To improve training stability, max-min normalization is adopted to transform different features into a similar scale. Then, the normalized features are connected to the input neurons in a one-to-one manner. The hidden layer embedded between the input and output layers is fully connected to these two layers and is the same size as the input layer. ReLU is used as the activation function for each neuron in the hidden layer. The activation function of the output neuron is a sigmoid function to determine whether there is a link from one node to another or not. This is a supervised learning model that learns how to map the input to the output, *i.e.*, the link features to the link label. The stochastic gradient descent (SGD) algorithm is used throughout the training process to minimize binary cross-entropy loss. After obtaining the trained model, the updated link features for the following year are fed into the model to predict its network topology.

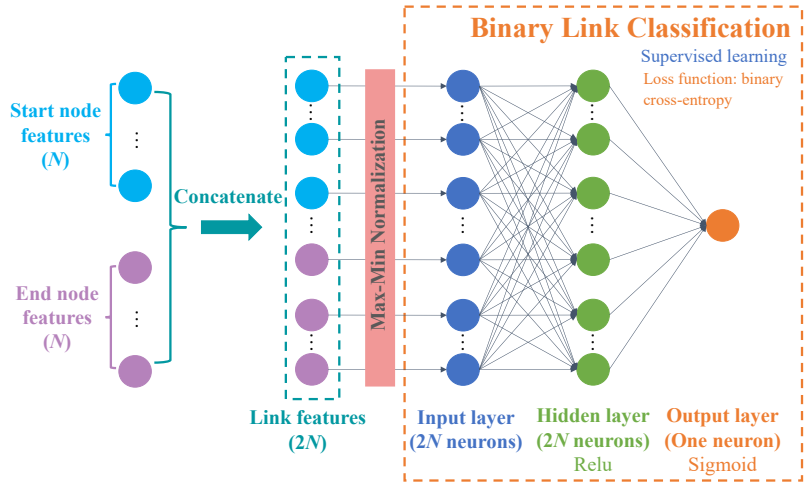


Figure 4.2: Architecture of the ANN model for link prediction.

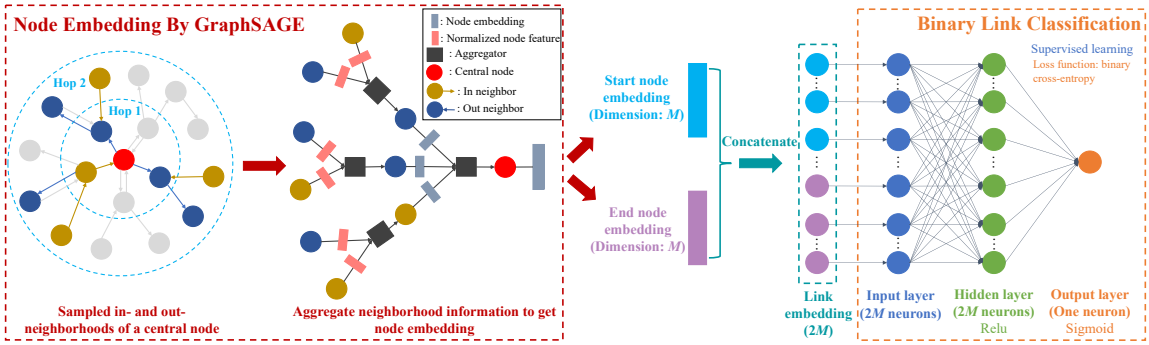


Figure 4.3: Architecture of the GraphSAGE model for link prediction.

4.2.3 The GNN-Based Link Prediction Model

4.2.3.1 Model architecture

As illustrated in Figure 4.3, a GraphSAGE link prediction model comprises two major parts: node embedding and link prediction. The node embedding is to learn a representation for each node in a vector of size M. Given a central node and its two-hop neighborhood, we first randomly sample its direct in- and out-neighbors at the first hop. Then, the same procedure is repeated to the sampled hop-1 neighbors to get their hop-1 in- and out-neighbors, *i.e.*, hop-2 neighbors of the central node. After that, the node features of hop-2 neighbors are normalized by max-min normalization

and used as the representations of hop-1 neighbors. Lastly, the node embedding of the central node can be obtained by tracing from hop-2 neighbors and aggregating their embeddings to hop-1 nodes and then to the central nodes inversely. The aggregator used in this study is the mean aggregator, where the node embeddings are computed by averaging neighboring node features (Hamilton et al., 2017).

Similarly to the ANN model, the learned node embeddings of the start node and end node are concatenated to represent the link embedding with size $2M$. Because the data have already been normalized during embedding, the link embedding is connected directly to the input layer, which is followed by one hidden layer and one output layer. The size of the input and hidden layers is the same as that of the link embedding. Other settings are identical to the ANN model for a fair comparison. In contrast to the ANN model that uses node features as input, node embedding learns information about a node's neighbors in addition to its own features in the network.

4.2.3.2 Model training and evaluation

During the training process, two types of data are fed into the model. One is the network data including node features and network adjacency matrix. The other one is the labels of all candidate links in the network, where existing links are labeled as class 1 and non-existing ones as class 0. The network data for GraphSAGE is to learn node embeddings, while the label data is for the learning task in link classification. This entire procedure is an end-to-end training to minimize the binary cross-entropy loss function by SGD (Ahmed et al., 2021).

When using testing data from the next year to evaluate the trained model's predictive performance, our input consists of the network data, including the updated node list and node features as well as the approximate network adjacency matrix. This approximate adjacency matrix is critical to have a correct link prediction by better estimating the embedding of a node in the future year.

4.2.3.3 Methods for adjacency matrix approximation

GraphSAGE assumes that if an embedding generating function of one type of network is learned, it can be used to generate node embedding by the same type of network. The assumption is that the training and testing networks should be of the same domain and have similar characteristics. It should be noted that since the new nodes (without any neighborhood information) do not have neighbors, GraphSAGE cannot use the adjacency information to make predictions for these nodes. In this study, since the shared mobility networks for training and testing are of the same month but in different years, they share similar characteristics. However, the challenge is that the test network for a future year is unknown. To obtain the embedding of the testing nodes, an approximate adjacency matrix must be obtained to estimate their neighbors.

According to the study (Ahmed et al., 2022a), there are several approaches to approximating the adjacency matrix, including directly using the training network or building a separate machine learning model for such an approximation. In our previous work (Yinshuang et al., 2022), we tested three methods.

1) The first method uses the adjacency matrix of the Period Two network obtained from the ANN model as input for the node embedding generation.

2) The second method employs a modified Period One mobility network. In this method, for those stations retained from Period One work, their neighbors are copied directly into the Period Two adjacency matrix. For the stations removed from Period One network, thus do not present in Period Two network, they are ignored. For those stations newly introduced in the Period Two network, they are kept independent and no neighborhood information is included in the embedding.

3) Finally, we use the real Period Two network to learn the node embedding and take its prediction performance as the ground truth to compare with the other

two approximation methods.

Based on the comparison results, we found that adopting the modified Period One mobility network to learn the node embedding of Period Two can generate the best prediction (the area under the precision-recall curve of the second method is much closer to the ground truth and exhibits 6-10% improvements compared to the first method). The adjacency matrix approximation by modifying the Period One network is thereby followed in this study.

4.2.4 Link Prediction Evaluation

Since link prediction in this study can be considered equivalent to binary classification, common metrics for binary classification can be adopted, including the confusion matrix, F1-Score, receiver operating characteristic (ROC) curve, precision-recall (PR) curve, and the area under the ROC and PR curves (a.k.a. AUC, area under curves). ROC AUC has a value between 0.5 (no skill) and 1.0 (perfect prediction); while PR AUC has a value between k (no skill) and 1.0 (perfect prediction), where k is the area under the no-skill PR curve, equal to the ratio of minority examples (class 1 links in our case) in the dataset. A higher AUC value indicates better predictive performance. For imbalanced classification problems where the majority of observations are negative cases and the minority of observations are positive cases, ROC analysis provides equal insights on the model's predictive performance in both cases. PR analysis focuses more on the model's ability to predict the minority case, *i.e.*, the positive links in the networks under current investigation (Brownlee, 2020).

4.3 Case Study: Divvy Bike in Chicago

In this section, we take Divvy Bike in Chicago as an example to demonstrate the utility of the proposed GNN-based models for shared mobility networks. In Section 4.3.2, the GraphSAGE link prediction model is compared with the ANN model

to test whether local network information (*i.e.*, node embedding features) impacts link prediction. In Section 4.3.3, we verify the generalizability of the proposed models by setting different link cutoffs.

4.3.1 Data Source

The Divvy Bike data for May 2016, referred to as Period One data, and May 2017, referred to as Period Two data, are used in this study (Divvy_Bike, 2020). We follow the approach described in Section 3.3 to process the data and build binary-directed trip networks by removing the links with less frequent trips (that is, those that occurred no more than four times in a month). Taking the Period One network as an example, a visualization of this binary-directed network is shown in Figure 4.4. The top hub stations' information for both two periods is listed in Table 4.1. The top five hub stations in Period One are observed to be the hubs in Period Two despite a slight change in ranking. In terms of POI information, a total of 2,269 POIs in Period One and 2,403 POIs in Period Two are collected based on the method presented in Section 4.2.1. According to Section 4.2.1, each station has geographic coordinates (two features), the number of docks (one feature), and POIs (seven features), for a total of ten features.

Table 4.1: Top five hub stations information in the trip networks of Period One (May 2016) and Period Two (May 2017)

Period One			Period Two		
Station ID	**Station Name**	**# of Connections**	**Station ID**	**Station Name**	**# of Connections**
287	Franklin St & Monroe St	320	77	Clinton St & Madison St	326
268	Lake Shore Dr & North Blvd	319	287	Franklin St & Monroe St	307
35	Streeter Dr & Grand Ave	317	35	Streeter Dr & Grand Ave	302
77	Clinton St & Madison St	316	91	Clinton St & Washington Blvd	295
91	Clinton St & Washington Blvd	303	268	Lake Shore Dr & North Blvd	295

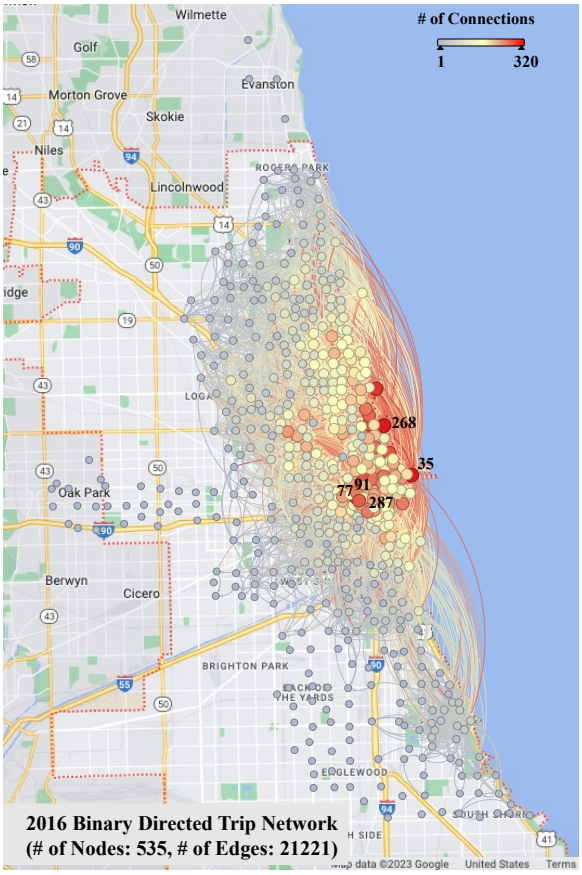


Figure 4.4: A visualization of the Divvy Bike trip network in May 2016. The nodes represent docked bike stations, and the directed links are trips that occur from one station to another with a frequency of more than 3 times in a month.

4.3.2 GraphSAGE-Based Link Prediction

4.3.2.1 Data preparation for ANN-based link prediction

In the ANN model, each candidate link within a trip network, represented by a pair of nodes, is a data sample. Consequently, there are 285,690 data samples in the Period One network, with 21,221 links classified as class 1 and 264,469 links as class 0. To reduce variability, the K-fold cross-validation approach (Bengio and Grandvalet, 2003) is applied, where K is set to 5. Therefore, we evenly split all class 1 links into five folds, *i.e.*, 20% data for each fold. Meanwhile, to avoid imbalanced training, the same number of class 0 links is randomly drawn without repetition from the class 0

sample pool and added to each fold. With these treatments, there are around 4,244 class 1 samples and 4,244 class 0 samples in one fold. Then, the cross-validation process alternately retains the first to the fifth fold for validation and the remaining four folds for training. In terms of the testing dataset, given that Divvy Bike had 582 stations during Period Two (May 2017), the total number of potential links from each pair of nodes in the testing dataset is 338,142. The final result is reported by averaging the K prediction results.

4.3.2.2 Data preparation for GraphSAGE predictive model

In addition to the data discussed above, the GraphSAGE model also requires network data to learn the node embeddings. For the training model of node embeddings, we feed it with the entire Period One network. As stated in Section 4.2.3.3, due to the fact that the Period Two network is unknown from a prediction point of view, we take a modified Period One network as input for the node embedding of the Period Two prediction. For those stations that are no longer operated in Period Two, they are removed and 48 new stations are added as isolated nodes. To ensure a fair comparison, we stick to the same configuration in the second stage for training the link classification model as was adopted in the ANN model.

4.3.2.3 Experiment settings

We first employ Bayesian optimization (Nogueira, 2014–) to perform hyperparameter tuning. The parameters that need to be optimized and their tuning ranges are listed in Table 4.2. In addition, we specify the objective of Bayesian optimization to minimize validation loss and set the training stopping criterion as 'no improvement in 10 epochs.' The number of tuning iterations is set to 15, the first five of which are random explorations. To reduce computational expenses, only the fold-one data is used to probe the best combination of hyperparameter values. The remaining four folds of the data are trained by following the same parameter settings.

Table 4.2: Hyperparameter tuning settings

Setting Items	Tuning Range
Minibatch size	[32, 240]
Learning rate	[1e-4, 1e-3]
Number of sampled in- and out-neighbors in two hops[1]	[5, 50]
Node embedding size	[10, 50]

[1]: Note that the numbers of sampled in-neighbors and out-neighbors can be different, we set them the same to simplify the model.

The results of hyperparameter tuning, as well as other hyperparameter values, are summarized in Table 4.3. Note that the epoch number of the optimized tuning results is 103. We extended it to 150 epochs for each training to further ensure the reliability of the training process. Finally, we ran experiments on single a machine with one NVIDIA P2200 GPU (5GB of RAM at 10Gbps speed), one 11th Gen Intel Core CPU (i9-11900 2.50GHz), and 32GB of RAM.

Table 4.3: Experiment parameter settings

Setting Items	Model Applied	Value
Neighborhood search depth	GraphSAGE	2
# of Sampled in- and out- neighbors in two hops		26
Node embedding size		26
Input and hidden layer size for GraphSAGE		52
Input and hidden layer size for ANN	ANN	20
Minibatch size	GraphSAGE and ANN	116
Epoch		150
Learning rate		3.49e-4
Dropout		0

4.3.2.4 Results for GraphSAGE-based link prediction

We first assess the performance of these two models using the confusion matrix and F1-Score, as shown in Table 4.4[2]. The left-hand side shows the confusion matrix

[2]The training time for ANN and GraphSAGE are 7 minutes and 35 hours, respectively. The primary factor affecting the computational efficiency of GraphSAGE is the process of in- and out-neighborhood sampling (Hamilton et al., 2017). However, since our proposed predictive model is

and the F1-Score of the ANN model. The confusion matrix includes four different combinations of predicted and actual classes, where there are 271,108 ± 1,830 true negatives, 47,177 ± 1,830 false positives, 914 ± 89 false negatives, and 18,943 ± 89 true positives. The true negative rate (TNR) and the false positive rate (FPR) reveal that 85.18% ± 0.58% of links in class 0 are predicted correctly, while 14.82% ± 0.58% are not. Similarly, the true positive rate (TPR) and the false negative rate (FNR) indicate that 95.40% ± 0.45% of the links in class 1 are correctly predicted and 4.60% ± 0.45% are not.

Table 4.4: Confusion matrices of Period Two link prediction via the ANN and GraphSAGE models (probability threshold = 0.50)

		ANN Link Prediction	
		0	1
Actual	0	271108±1830 (TNR 85.18% ± 0.58%)	47177±1830 (FPR 14.82% ± 0.58%)
	1	914±89 (FNR 4.60% ± 0.45%)	18943±89 (TPR 95.40% ± 0.45%)
F1-Score		0.44 ± 0.01	
		GraphSAGE Link Prediction	
		0	1
Actual	0	290619±1164 (TNR 91.31% ± 0.37%)	27666±1164 (FPR 8.69% ± 0.37%)
	1	2259±122 (FNR 11.38% ± 0.61%)	17598±122 (TPR 88.62% ± 0.61%)
F1-Score		0.54 ± 0.01	

Similar results of the GraphSAGE model are listed on the right-hand side. We observe from these two matrices that the ANN model shows a more accurate true positive prediction where the TPR is around 7% higher than that of the GraphSAGE model when taking 0.50 as the probability threshold. However, this outperformance is offset by the higher true negative prediction of GraphSAGE, which is approximately 6% higher than that of the ANN model. The same conclusion can be reached by

not meant for real-time forecasting, we, therefore, prioritize improving model performance even if it means sacrificing computational efficiency.

comparing their F1-Scores, which show that the F1-Score of GraphSAGE is 0.1 higher than the ANN model.

We then compare these two models at the aggregated level by the ROC and PR AUCs. There are inconspicuous differences between the two ROC AUC values of the ANN and GraphSAGE models, both of which are equal to 0.96. Their high AUC value (greater than 0.95) indicates that these two models show identical and considerable performance when the predictions of the majority class and the minority class are treated equally important. However, the evident gap between the two PR curves shown in Figure 4.5 implies that the GraphSAGE model outperforms the ANN model when the minority class prediction is the focus, *i.e.*, whether the class 1 (positive) links are correctly predicted or not. The PR AUC of the GraphSAGE model is about 8% higher than that of the ANN model. This implies that the local network information aggregated by GraphSAGE can enhance the model's performance in the prediction of positive links that are more important to design decisions.

4.3.3 GrapSAGE-Based Link Prediction for Networks With Different Link Strengths

In Section 4.3.2, we set the link cutoff at 3.03, which is the mean minimum link weight of monthly travel networks throughout the year from 2014 to 2017, following the approach described in Section 3.3. To assess the generalizability of the proposed predictive model and demonstrate the importance of neighborhood information for different network sizes, we change the cutoff value from 0 to 16, corresponding to the ratio of positive links decreasing from 100% to 10%. Note that as the number of positive links declines, the data becomes even more imbalanced, with the majority of links being negative, thus making prediction even more challenging. Furthermore, to more easily trace the trend of prediction accuracy, we follow the experiment settings given in Table 4.3 and perform a five-fold cross-validation to train models with different link cut-off points.

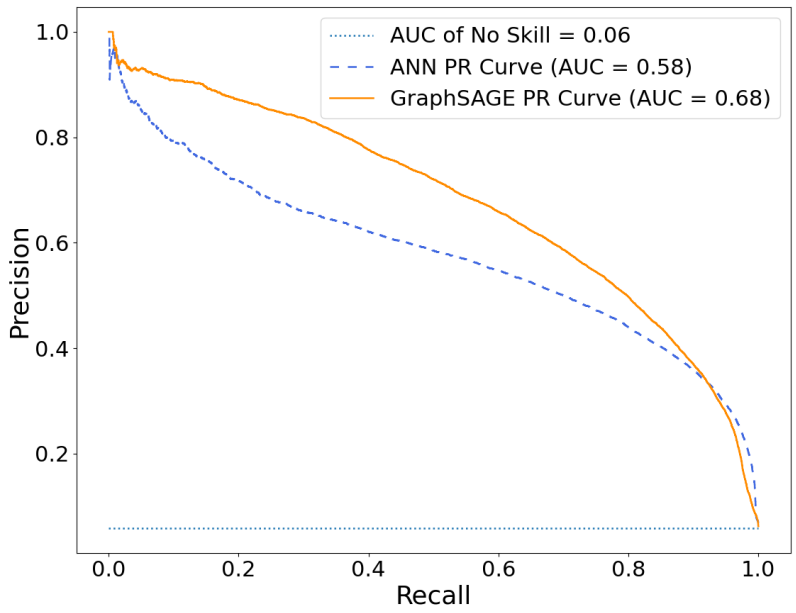


Figure 4.5: PR curve example of Period Two link prediction using the ANN and GraphSAGE predictive models in fold four. The average PR AUCs are 0.59 $\pm$ 0.01 and 0.67 where the GraphSAGE model has a higher AUC than the ANN model in the PR curve.

4.3.3.1 Results for link prediction for networks with different link strengths

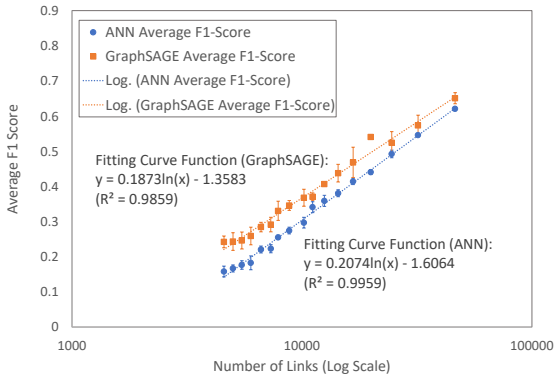
The evaluation metrics include F1-Score, ROC AUC, and PR AUC, all of which are averaged based on five-fold results. We first investigate the overall predictive performance of both models using F1-Score when the probability threshold is equal to 0.5, which is shown in Figure 4.6 (a). It is evident that the predictive powers of both GraphSAGE and ANN models decrease when the network becomes sparse, and the reason could be attributed to the aforementioned worse imbalanced issues of sparser networks. Furthermore, the consistently higher F1-Score of the GraphSAGE model suggests that neighborhood information indeed plays a role in improving prediction accuracy. The same conclusions can be drawn from the PR AUC plots in Figure 4.6 (b) when the emphasis is placed on the prediction of the minority class (positive links) across all probability thresholds.

In contrast, we find that regardless of network size, both ANN and GraphSAGE predictive models maintain the same high ROC AUC, which is around 0.96. This implies that these two models share a similar predictive power when putting the spotlight on both the majority and minority classes, and this power is robust enough to against the decline of the minority class.

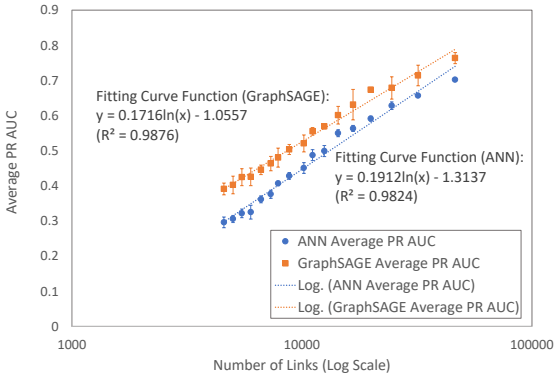
Lastly, to further validate that the decreased performance of GraphSAGE is irrelevant to the network topology, we conducted an experiment by using a subset of 2016 training data (showing totally different network topology) and 2017 testing data to test the model's predictive power. These synthetic data are generated by identifying the top 166 popular Divvy Bike stations in 2016 and the trips that occurred among these stations in 2016 and 2017. Although the trip networks constructed by these stations and trips exhibit distinct typologies in terms of their degree distributions compared to the real trip networks, the same decreased trend of predictive performance is still observed. Furthermore, the decreasing rate of the performance of both the synthetic network and the real network is highly correlated with the shrinking speed of the network size, thereby again demonstrating that the primary reason for poor prediction accuracy comes from the imbalance issue of sparse networks, rather than from network typologies.

4.4 Link Prediction (LP) Model to Support System Design Decision-Making

In this section, the proposed GraphSAGE model is utilized to assist SMS designers or other stakeholders in predicting system performance after a design strategy is proposed. We first formulate a design case and then evaluate the influence of the design decisions on users' trips and compare the prediction accuracy of GraphSAGE by comparing it with the baseline ANN model.



(a) Average F1-Score (with error bars, probability threshold = 0.5)



(b) Average PR AUC (with error bars)

Figure 4.6: F1-Scores and PR AUCs change with the number of links. The rightmost points in the plots correspond to 46,352 links when the cutoff value is equal to 0. We notice that the average F1-Scores and PR AUCs of both GraphSAGE and ANN models decrease logarithmically with the shrink of the network sizes, and the GraphSAGE model consistently has higher values than the ANN model.

4.4.1 Divvy Bike Design Case

There are two levels of design decisions in this system. The first is *capacity-level design decision* that a designer should determine, *i.e.*, the stations that need to be expanded or contracted and the number of docks that each station needs to add or remove. The second is *station-level design decision*, that is, a designer needs to decide (at a certain time point) which existing stations need to be removed and where new stations shall be built. By comparing the data from the Divvy bike station in May 2017 with the data from the Divvy bike station in May 2016, we assume that a decision maker proposed the following two-level design decisions at the end of May 2016 and would like to estimate their influence by predicting the connections of these key stations in May 2017.

1) *The capacity-level design decision*: stations in the set $S_1 = \{341, 195, 97, 72\}$ are planned to expand by adding 16 docks; stations in the set $S_2 = \{444, 496, 2, 445, 400, 489, 412, 407\}$ are planned to shrink by removing 8 docks. The

locations of the stations in sets S_1 and S_2 are marked with dark blue and light blue pins, respectively, in Figure 4.7.

2) *The station-level design decision*: station in set $S_3 = \{372\}$ is planned to remove; 48 new stations in set $S_4 = \{524, 578, 522, 622, 550, 531, 517, 581, 575, 585, 523, 584, 580, 525, 520, 576, 619, 590, 591, 592, 623, 589, 586, 526, 620, 515, 579, 582, 514, 588, 573, 583, 577, 571, 574, 587, 595, 405, 527, 519, 602, 603, 598, 604, 605, 599, 606, 612\}$ are planned to construct. The locations of the stations in sets S_3 and S_4 are marked with red and green pins, respectively, in Figure 4.7.

We update the May 2016 network by applying these proposed design decisions. For example, the capacity and location information designed for the new stations in the set S_4 is added to the station list of 2016. Regarding their POI data, we adopted the approach described in Section 4.2.1 to count the number of each type of POIs around these designed stations in 2016. With the updated 2016 network, we used the trained GraphSAGE and ANN models to predict the connections of these critical stations within sets S_1, S_2, and S_4, as well as evaluate the prediction by comparing them with real connections in 2017.

4.4.2 Capacity-Level Station Connection Prediction

Taking the expansion station set S_1 as an example, we assume that the stations in S_1 are connected with n out of the N stations in Period Two in reality. For example, these stations connect $n = 149$ stations in 2017 and Divvy Bike had $N = 582$ in total that year. The GraphSAGE model predicts that these stations connect with M stations when the probability threshold is p, and m stations are correctly predicted. Therefore, the true positive (TP) equals m, the false negative (FN) equals $n - m$, the false positive (FP) equals $M - m$, and the true negative (TN) equals $N + m - n - M$. The PR curves correspondingly obtained across all probability thresholds from 0 to 1.

Figure 4.7: The geographical locations of stations in sets S_1, S_2, S_3 and S_4.

The confusion matrices and F1 scores of the expansion stations in S_1 and contraction stations in S_2 when the probability threshold is 0.50 are presented in Table 4.5 and Table 4.6. The results in Table 4.5 indicate that ANN more accurately predicts expansion stations' connections with higher TPR (98.93%) than GraphSAGE (TPR=93.83%), but greatly sacrifices TNR (64.53%). In the contraction case, t-tests are conducted, comparing the means of ANN and GraphSAGE in terms of their TNRs, TPRs, and F1 scores. The null hypothesis is that there is no difference. The resulting *p-value*, 0.01, denotes a significant difference between the means of the ANN and the GraphSAGE TPRs, indicating that the GraphSAGE TPR (95.96%) is higher than that of the ANN (92.77%). However, the *p-values* of 0.23 and 0.07 in the tests of

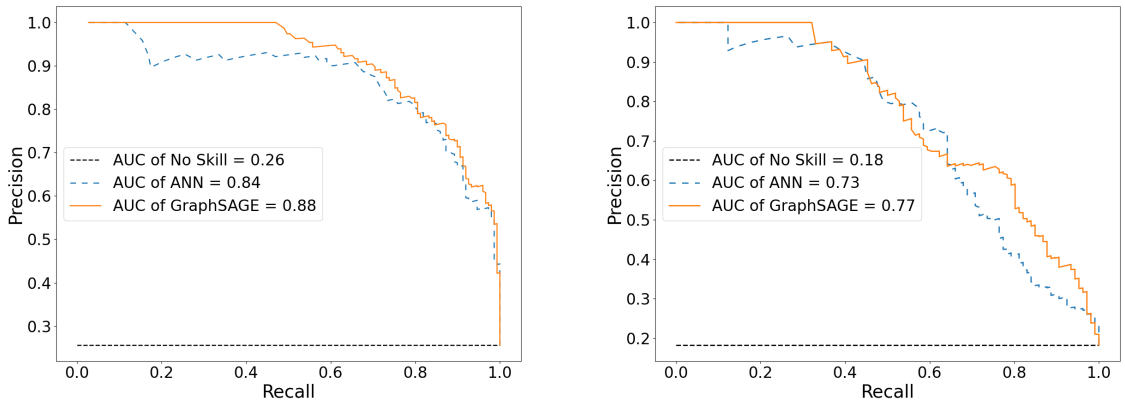
Table 4.5: Confusion matrices of expansion station trip network connections via ANN and GraphSAGE predictive model (probability threshold = 0.50). "Not Connection" denotes stations that were not connected to the stations in the set S_1 by trips in 2017 and vice versa for the "Connection" term. Similar definitions apply to Table 4.6 and Table 4.7.

	ANN Prediction	
	Not Connection	Connection
Not Connection	279±12 (TNR 64.53%±2.81%)	154±12 (FPR 35.47%±2.81%)
Connection	2 (FNR 1.07%±0.33%)	147 (TPR 98.93%±0.33%)
F1-Score	0.66±0.02	
	GraphSAGE Prediction	
	Not Connection	Connection
Not Connection	365±3 (TNR 84.20%±0.69%)	68±3 (FPR 15.80%±0.69%)
Connection	9±2 (FNR 6.17%±1.61%)	140±2 (TPR 93.83%±1.61%)
F1-Score	0.78±0.01	

Table 4.6: Confusion matrices of contraction station trip network Connections via ANN and GraphSAGE predictive model (probability threshold = 0.50).

	ANN Prediction	
	Not Connection	Connection
Not Connection	316±21 (TNR 64.84%±4.21%)	172±21 (FPR 35.16%±4.21%)
Connection	7±1 (FNR 7.23%±1.56%)	87±1 (TPR 92.77%±1.56%)
F1-Score	0.50±0.03	
	GraphSAGE Prediction	
	Not Connection	Connection
Not Connection	331±8 (TNR 67.75%±1.62%)	157±8 (FPR 32.25%±1.62%)
Connection	4±1 (FNR 4.04%±1.04%)	90±1 (TPR 95.96%±1.04%)
F1-Score	0.53±0.01	

TNR and F1-Score imply that ANN and GraphSAGE perform an identical predictive power in the contraction case when the probability threshold = 0.50.



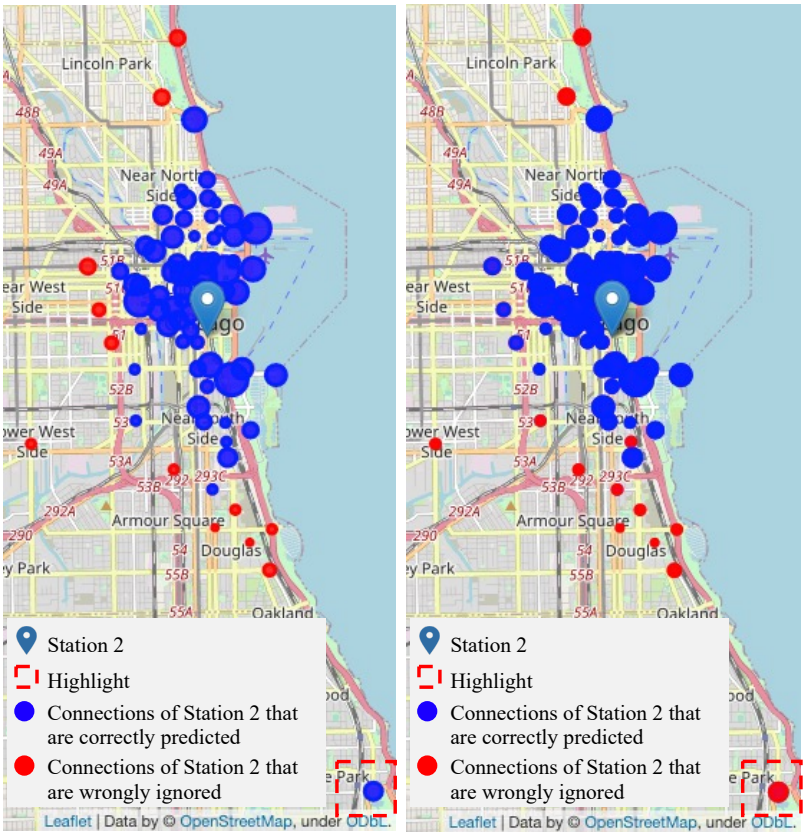
(a) Expansion Case

(b) Contraction Case

Figure 4.8: PR curve example of *capacity-level design decision* evaluation through four-fold ANN and GraphSAGE trained models by predicting the network connections of key stations. We observe that the AUCs of the GraphSAGE model are 3% $\sim$ 5% higher than the ANN model in both expansion and contraction cases. For the expansion case, the average PR AUC of the Graphsage model is 0.88, which is 3% higher than that of the ANN model, equal to 0.85 ± 0.01. In terms of the contraction case, the average PR AUCs of the Graphsage and ANN models are, respectively, 0.78 ± 0.01 and 0.73 ± 0.02.

Overall, the F1-Scores of GraphSAGE on both expansion and contraction cases show its superiority, implying the important role of neighborhood information in influencing users' behaviors in BSS. This conclusion is further validated by the PR curves shown in Figure 4.8. That being said, when designers are to evaluate their proposed capacity-level design strategies, the GNN-based model is more reliable.

To visually demonstrate the models' predictive performance in this regard, we take Station 2 in S_2 as an example, as shown in Figure 4.9. The graphs show a decent predictive performance where over 80% connections of Station 2 are correctly predicted. Furthermore, GraphSAGE, which considers neighborhood information, slightly improves the prediction accuracy and correctly predicts the geographically farthest connection of Station 2, which is located in the lower right corner of the plots (highlighted by the red dashed square).



(a) GraphSAGE (b) ANN

Figure 4.9: Link prediction of contracted design case, Station 2, using the GraphSAGE and the ANN predictive model. The size of the dots depicts the capacity of the station. (a) is the GraphSAGE predicted result when the probability threshold is equal to 0.78, where 0.78 is the optimal threshold for the GraphSAGE PR curve in Figure 4.8 (b). 68 of the 80 connections (85.00%) of Station 2 are correctly identified. (b) is the ANN predicted result when the probability threshold is equal to 0.88, where 0.88 is the optimal threshold for the ANN PR curve in Figure 4.8 (b). 67 of 80 connections (83.75%) of Station 2 are correctly predicted. For the selection of optimal thresholds, please refer to our previous work (Yinshuang et al., 2022).

4.4.3 Station-Level Station Connection Prediction

With regard to predicting the connections of the newly built stations in S_4, we tested one ANN model and two different GNN models, as shown in Figure 4.10. The distinction between GraphSAGE and GraphSAGE-GroundTruth is that GraphSAGE kept the newly built stations independent and did not include their neighborhood

information in the embedding. GraphSAGE-GroundTruth, instead, took the real neighborhood information from the new stations in May 2017 into the construction of network embeddings to test the best scenario that GraphSAGE prediction can reach. The AUC results indicate that GraphSAGE shows no better predictive power than ANN for these isolated new stations when there is no neighborhood information input. This is validated by the higher AUC of the GraphSAGE-GroundTruth model and its F1-Score in Table 4.7.

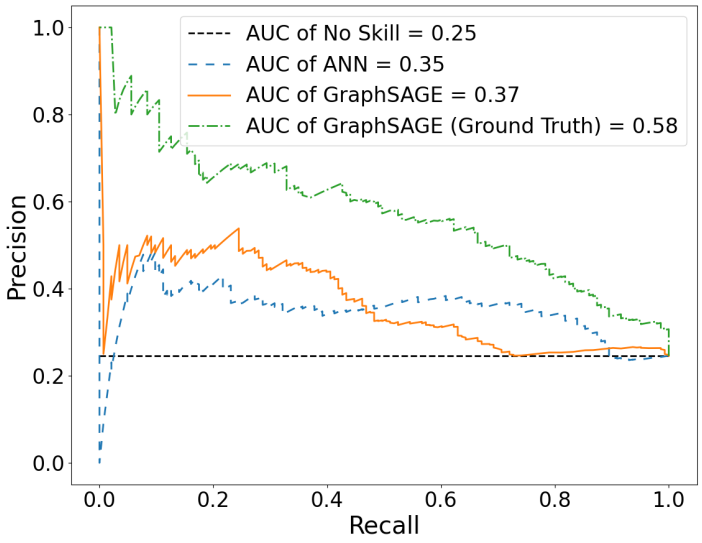


Figure 4.10: PR curve example of *station-level design decision* evaluation via the fourth fold ANN and GraphSAGE trained models by predicting the network connections of the key stations. The average PR AUCs of ANN, GraphSAGE, and GraphSAGE (Ground Truth) by five folds are 0.33 ± 0.02, 0.36 ± 0.02, and 0.55 ± 0.04.

4.5 Discussion

There are a few limitations of the proposed GNN-based design decision support model in the current study that can lead to future investigations. First, the relatively lower predictive performance for the BSS network with a smaller size, having a worse imbalance issue, shows that our proposed model is better suited for link prediction

Table 4.7: Confusion matrices of newly built station trip network connections via ANN and GraphSAGE predictive model with ground truth (probability threshold = 0.50).

	ANN Prediction	
	Not Connection	Connection
Not Connection	163±14 (TNR 37.13%±3.29%)	276±14 (FPR 62.87%±3.29%)
Connection	24±5 (FNR 16.92%±3.47%)	119±5 (TPR 83.08%±3.47%)
F1-Score	0.44±0.01	
	GraphSAGE Prediction (Ground Truth)	
	Not Connection	Connection
Not Connection	245±7 (TNR 55.72%±1.57%)	194±7 (FPR 44.28%±1.57%)
Connection	18±2 (FNR 12.73%±1.28%)	125±2 (TPR 87.27%±1.28%)
F1-Score	0.54±0.01	

of a denser network. It should be worthwhile to study methods that can address this imbalance challenge and thus broaden the application of the model.

Second, a few assumptions made in this work could potentially impede the model from capturing reality, thus weakening its validity. For example, in the design case study given in Section 4.4, we assumed that all design decisions are made at the same time, *e.g.*, at the end of May 2016. However, in reality, the decisions could be scattered across different months, so a dynamic model is preferred in this scenario to predict the station linkage in the short term. Also, we observed that it was rare that a user rented a bike and then returned it to the same location. But, our model is not designed to predict rare self-loop links. For example, in a mini-experiment, we selected a station (station 30) at random and created two duplicate stations (1030 and 1031) with identical attributes, such as the same dock numbers, locations, and surrounding POIs, as well as the same network neighbors. We then used the trained GraphSAGE model to predict the link probabilities of stations 30, 1030, and 1031 connecting to other stations separately. According to the results, we observed that there is a high probability of forming connections among the three duplicated stations, leading to

the dominance of self-loop trips. Therefore, when using the model in a situation where there are duplicated nodes, it is better to assume that self-loop trips are not allowed. Another strategy is to combine all duplicated nodes in one node by stacking the number of docks of each node.

Third, the accuracy in predicting the newly added stations is relatively low, as shown in Section 4.4.3. To make this model applicable to the new stations, efforts are required to test more adjacency matrix approximation approaches and, subsequently, better estimates of the network neighborhood information of newly introduced stations. When applying the proposed model for decision support, it is more suitable to predict connections and travel demand between stations that have already been on the network.

Finally, in this study, only a few node features (station capacity, geographic coordinates, and surrounding POIs) are considered. However, there could be other factors influencing the accuracy of predictions, such as unserved travel demands (*e.g.*, instances where people attempted to rent a bike at an empty station). One key reason for missing this information is the availability and accessibility of the data. For example, obtaining data on failed rental attempts at an empty station, which is essential to capture unserved travel demands, is not readily available in the current dataset. However, our proposed model is adaptable and generalizable to incorporate additional features. In our future study, a potential way to address this data limitation is to conduct surveys or statistics from BSS service Apps to estimate unserved travel demands.

4.6 Conclusion

In this chapter, we present a complex network-based prediction framework for STS design support with graph neural networks. Taking SMS as the case study, we utilize this approach to predict whether two stations in an SMS would have sufficient travel demand to form a connection over a long timescale. The utility of the proposed

approach in supporting system decisions in shared mobility networks is investigated and validated. In particular, we examine whether local network information impacts link formation using GNN models. In the case of Divvy Bike in Chicago, two-hop neighborhood information is used to generate node embeddings for link prediction. The results show that the GNN model with local network information outperforms the one without, revealing the important role of local network structures in the formation of trip networks at the system level. We also test the model performance using local network information by changing the link strength from weak to strong, corresponding to the network size from large to small. The results indicate that the GNN model has maintained better performance than the ANN model regardless of the imbalanced data issue. The knowledge generated by these analyses contributes to answer ***RQ2***

Finally, we present a design case study to illustrate how to apply the GNN-based predictive model to assist system designers in gaining insights into their proposed design decisions. Using the trained GNN model to predict the network neighbors of the expansion stations, the contraction stations, as well as the newly built stations, we demonstrate the applicability of the predictive model in helping system designers make an initial assessment of the network connections of these critical stations. However, despite the fact that the current prediction of new stations is not satisfactory, the GroudTruth result validates that improvement can still be achieved once a better approximation of the adjacency matrix is obtained. This illustration of using the predictive model to assist STS design contributes to answer ***RQ3***.

Chapter 5: Information Retrieval and Survey Design for Networked Socio-Technical System Data Collection[1]

5.1 Overview

The scarcity of data poses a persistent challenge in the engineering and design of socio-technical systems (STSs). In response, this chapter proposes the application of information retrieval and survey design methods for collecting networked STS data. Two distinct approaches are presented to illustrate different data collection processes. Firstly, a combination of web-crawling methods and survey questionnaire design is employed to gather data on the US household vacuum cleaner market system. Additionally, a survey web platform is developed to facilitate the survey study. Secondly, we employ a blend of web-crawling techniques and named entity recognition (NER) models, a subset of natural language processing (NLP) models, to extract network data pertaining to the US vehicle market system from text gathered from social media platforms. This chapter serves as a foundational element in the engineering and design of STSs, with the collected data being frequently utilized in subsequent chapters.

The chapter is organized as follows:

- Section 5.2 introduces a framework for product co-consideration network data collection based on survey design. The US household vacuum cleaner is taken as a case study for illustration.
- Section 5.3 provides an approach for product co-mentioning network data collection based on social media mining. The US vehicle is taken as a case study

[1]The content of Section 5.2 has been published in (Xiao et al., 2024). My contributions include conceptualization, methodology, formal analysis, visualization, and article writing. The content of Section 5.3 has been published in (Gavino et al., 2023). My contributions include conceptualization, methodology, and article writing.

for illustration.

- Section 5.4 concludes this chapter with closing thoughts.

5.2 US Household Vacuum Cleaner Market Network Data Collection

An overview of the data collection process is shown in Figure 5.1. It consists of four major steps, each described in detail below.

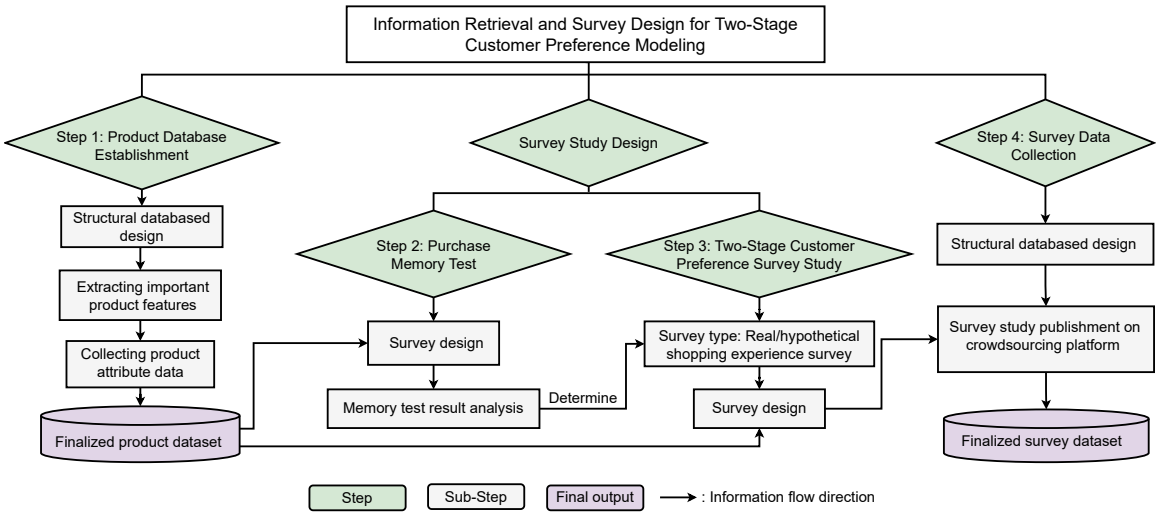


Figure 5.1: An overview of the vacuum cleaner market network data collection process.

5.2.1 Step 1: Product Database Establishment

To start the process, we collected information on household vacuum cleaners using two web crawling techniques – Beautiful Soup and Selenium in Python. Five primary categories of vacuum cleaner data, *i.e.*, upright, canister, stick, handheld, and robotic vacuum cleaners were obtained from mainstream online shopping platforms in the US market, including Amazon, Wayfair, Best Buy, Home Depot, and Walmart. After web scraping, data was cleaned to merge data from different sources; meanwhile,

the duplicated data and noises were removed too. In the end, 1170 vacuum cleaner products were collected. The collected information includes product title, product image, product model name, SKU (stock-keeping unit), product description, customer rating, customer reviews, and 26 product features (list price, product dimension, weight, manufacture, brand, color, capacity, etc.).

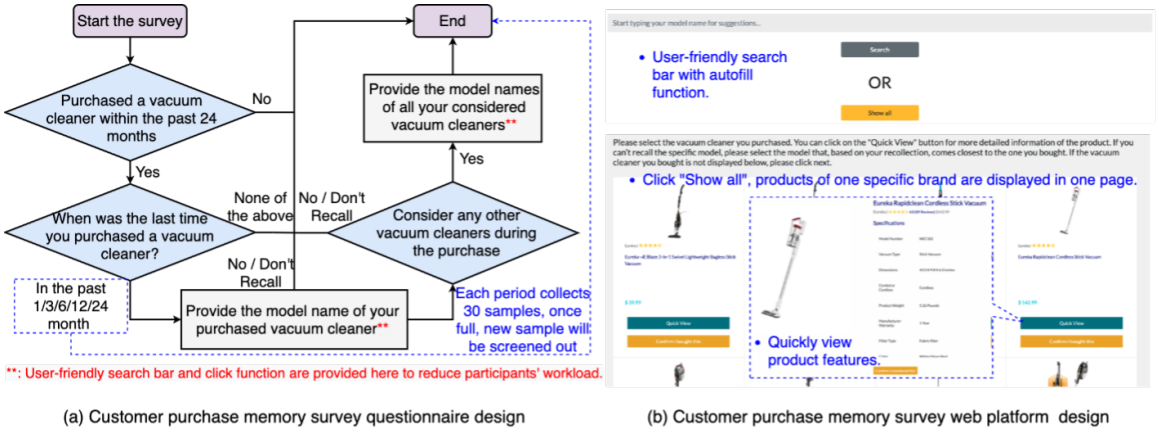
In addition, we extracted product features from online customer reviews to determine the most important (most frequently mentioned) features that shall be included in the survey questions. We scraped 60,000 reviews from Amazon (200 reviews for each product) and used a rule-based semi-supervised learning model (Rana and Cheah, 2017) for extracting features and sentiment/opinions associated with those features. For example, some feature-opinion pairs extracted from the reviews include "strong suction," "heavy weight", "annoying cord," and "loud noise." After obtaining candidate features from the opinion mining, unrelated features were pruned. The remaining features were then ranked based on their frequency in customer reviews (Rai, 2012). In the end, we identified 22 important product features based on the results from opinion mining, including attributes such as price, product type, floor surface recommendation, suitable for pet hair, suction power, noise, power source, bag or bagless, cord or cordless, battery charge time, HEPA filter, warranty, brand, color, weight, dimensions, power, capacity, overall customer ratings, and three robotic vacuum cleaner specific attributes (navigation system, voice control, and remote controls).

5.2.2 Step 2: Customer Purchase Memory Test

To ensure the credibility of the two-stage customer preference survey study, a memory test was conducted to evaluate customers' abilities to recall their decision-making process while purchasing vacuum cleaners in five different periods: one month, three months, six months, twelve months, and 24 months. This helped us determine the type of survey (*i.e.*, real or hypothetical shopping experience survey) and the ap-

propriate threshold for soliciting participants. In the real one, the survey will be conducted only among the participants who actually purchased the product. In the hypothetical one, participants will be required to complete a survey based on a virtual online shopping experience.

As illustrated in Figure 5.2 (b), an online survey web was designed and developed. The survey web connected with the product database generated in Step 1 to create a simulated online shopping system. Additionally, we designed user-friendly interfaces, such as the product search bar and product preview, to facilitate participants in identifying the vacuum cleaners they considered and purchased. We collected 30 respondent samples for each period and calculated the proportion of participants who could recall the specific models they considered and purchased. If the proportion exceeded 50%, we considered the customers' memory within that time period to be reliable.



(a) Customer purchase memory survey questionnaire design

(b) Customer purchase memory survey web platform design

Figure 5.2: Survey questionnaire flowchart and web platform design for customer purchase memory test (Xiao et al., 2022b).

The pilot survey study was conducted on the Cint platform from December 18 to December 21, 2020. Table 5.1 summarizes the actual collected sample size for the test. It was noted that there were significantly fewer samples for the 24-month scenario than for the other periods, so this scenario was excluded from the proportion calculation. Figure 5.3 indicates that 62% of customers who purchased a

Table 5.1: Sample sizes for the purchase memory test (Xiao et al., 2022b).

	In the past 1 month	**In the past 3 months**	**In the past 6 months**	**In the past 12 months**	**In the past 24 months**
# of respondents who purchased a vacuum cleaner	32	34*	32	35	8

*: This number has excluded the number of people who have purchased a vacuum cleaner in the past one month. A similar operation was applied to the other three periods (in the past 6/12/24 months).

vacuum cleaner in the past three months can recall their purchases and considerations, meeting the 50% threshold. However, focusing solely on customers who purchased vacuum cleaners within the past three months may not yield enough samples for the subsequent two-stage customer preference survey in Step 3. To strike a balance, the survey study in Step 3 was extended to include customers who made purchases within the past six months as they had a high recall ratio for purchase (75%) and an acceptable ratio for a recall of both purchase and consideration (43.75%). Therefore, according to the memory test results, we decided to conduct a study on customers' revealed preferences and recruited participants who had purchased a vacuum cleaner in the past six months.

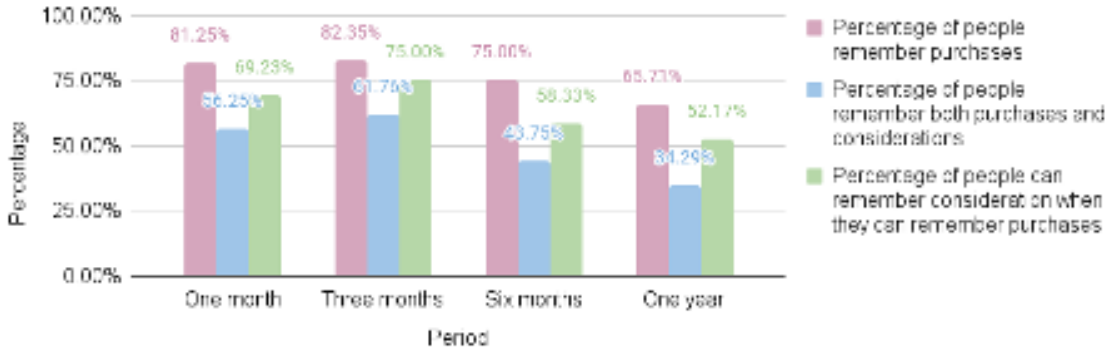


Figure 5.3: The ratio of participants who can recall the purchased or considered vacuum cleaners (Xiao et al., 2022b).

5.2.3 Step 3: Two-Stage Customer Preference Survey Questionnaire Design

Step 3 involves designing the two-stage customer preference survey questionnaire. As shown in Figure 10, the questionnaire consists of four major parts. Part One includes two filtering questions to collect respondents' vacuum cleaner purchase decisions, which are the most important information we want to collect. Only the respondents who purchased a vacuum cleaner within the past six months and could recall the products they purchased were allowed to participate in the rest survey.

In Part Two, the online survey web shown in Figure 8(b) was used to collect participants' historical consideration and choice data. They were asked to provide information about the type, brand, and exact models of vacuum cleaners they have considered and purchased, as well as the top-rated design attributes (product features) that influenced their choice-making. Participants could rank these attributes by dragging them from a list of features identified by the feature selection algorithm introduced in Step 1 to the corresponding text boxes.

In Part Three, we design questions to collect participants' social network data. This was relevant because social networks can influence consumers' purchase decisions. Participants were asked to provide information on their general social networks (GSN) as well as product-specific social networks (PSN), both of which have the potential to influence participants' choice behaviors. Each participant was asked to provide information for at least one and up to five individuals in their GSN with whom they discuss daily matters. Additionally, they were asked to provide information for up to five other individuals in their PSN with whom they had discussed the vacuum cleaner purchase. Therefore, each participant can nominate up to a total of ten different people in their social network for the study. These individuals' demographic data and their contact frequencies with the respondents were also recorded.

Part Four aimed to collect personal information and general preferences of the participants, such as their demographics and viewpoints about vacuum cleaners.

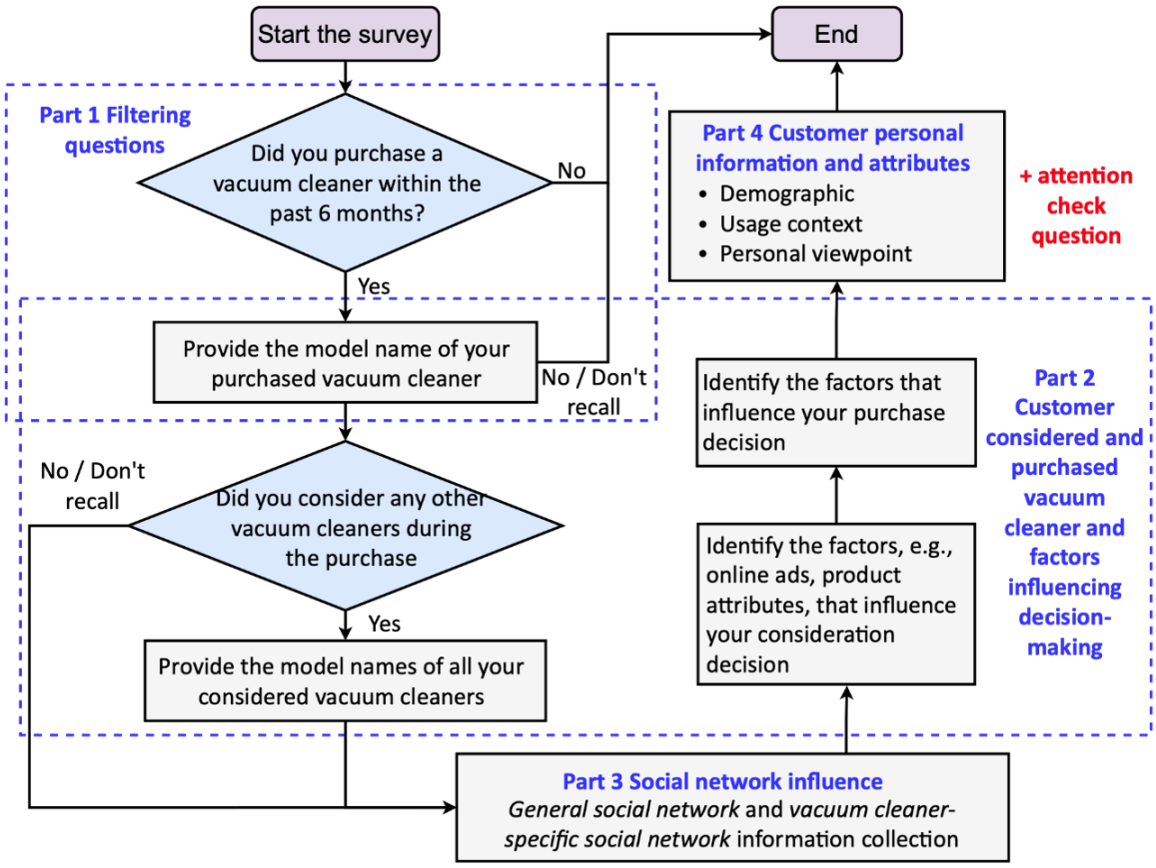


Figure 5.4: Two-stage customer preference survey questionnaire flowchart.

Additionally, this part of the survey focuses on understanding the product usage context of the participants, including how often they use the vacuum cleaner and where they use it. To ensure the quality of the survey data, we employed several strategies (Flowerdew and Martin, 2005):

- Designed and implemented attention check questions;
- Organized questions by placing important ones first and less important ones last;
- Made questions mandatory to avoid missing data, *i.e.*, participants could not proceed to the next stage unless answering all the required questions on the current page;

- Conducted both internal and external pilot studies to collect feedback on the questionnaire;
- Incorporated experts' inputs and feedback from multiple disciplines, including engineering design, social science, and psychological science.

5.2.4 Step 4: Survey Data Collection

We launched our survey on Cint, a digital insights gathering platform with quality assurance mechanisms such as artificial intelligence (AI)-driven fraud detection system. To ensure reliable data storage, the survey data was automatically saved in an SQL database on pgAdmin, with a structured column sequence. This database had been configured to communicate effectively with the survey website. To acquire more results, the survey was distributed to different groups, such as those who had recently purchased a vacuum cleaner or those who were interested in home decoration and home appliances. Meanwhile, to mirror the real market, a quota sampling technique [9] was used to match the age distribution of the US census. The survey was conducted over two months, from April 25 to June 25, 2021, with the aim of collecting approximately 1,000 complete responses. To improve the reliability of data collection, we divided this data collection process into four phases. Each phase targeted an equidistant increase, with goals set at 100, 200, 300, and 400 complete responses from Phase 1 to Phase 4. Table 2 provides a summary of the actual number of participants and the complete responses obtained in each phase. After obtaining a total of 1023 complete responses, a subsequent manual check identified 21 responses related to hard-to-find vacuum cleaners, prompting their removal. Finally, a total of 6585 participants attended the survey and 1002 complete responses were received, with a completion rate of 15.21%.

From the data collected, we identified 624 unique vacuum cleaner models that had been either considered or purchased by the respondents. However, given that the scrapped product attribute data in Step 1 included a considerable number of

Table 5.2: The total number of participants and the number of complete responses received in each phase. Participants' responses could be removed due to: 1) early screening: Participants who did not purchase a vacuum cleaner, disagreed with the survey agreement, or did not specify their purchased vacuum cleaners, were screened early in the process; 2) incomplete survey: Participants who did not complete the survey in its entirety were excluded; 3) attention check failures: participants who did not pass the attention check questions were excluded; 4) suspicious cheating: Instances of suspicious behavior, such as inputting irrelevant words or sentences in text boxes and consistently providing the same answer (*e.g.*, "Strong Agree") to all personal viewpoint questions, led to participant removal.

	Phase 1	**Phase 2**	**Phase 3**	**Phase 4**
# of Participants	828	1263	2002	2492
# of complete responses	101	220	292	410

missing values, we conducted an additional round of manual data collection to address the missing value issue. This manual collection involved gathering information from various sources, such as product specifications and manuals, the brand's official online stores, and expert performance review reports available online.

5.3 US Vehicle Market Network Data Collection

In this section, we use the US vehicle market as a case study to illustrate the framework that integrates information retrieval and named entity recognition (NER) techniques for gathering STS network data from unstructured social media sources. Figure 5.5 is an overview of the proposed framework.

5.3.1 Step 1: US Vehicle Attribute Data Collection

Similar to the vacuum cleaner case study, we initiate the process by gathering information from Car.com on mainstream vehicle models using Beautiful Soup and Selenium in Python. The collection occurs in two phases. Initially, spanning from 2010 to 2022, we compile a list of 949 unique car model names in English. Subsequently, in

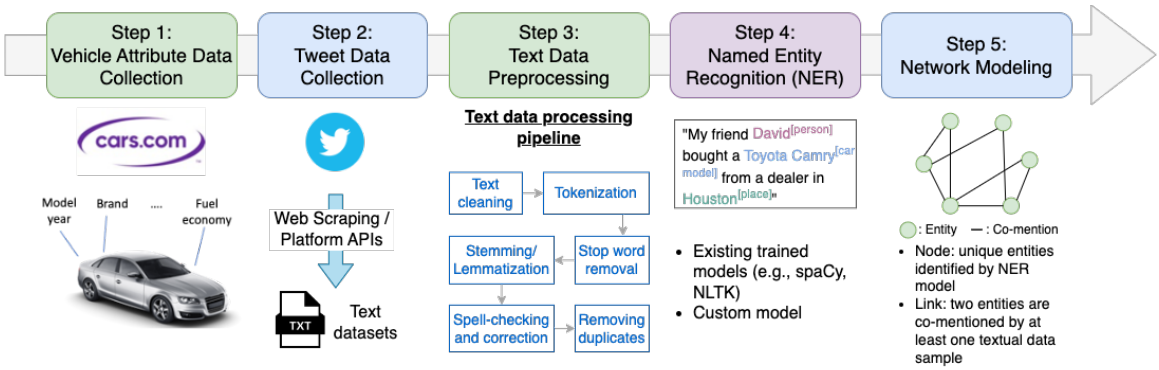


Figure 5.5: Framework of STS network data collection from social media.

the second phase, we treat car models from different years as distinct entities (*e.g.*, 2016 Honda Accord versus 2017 Honda Accord) and collect attribute data for each vehicle year ranging from 2016 to 2022. Following the acquisition of raw attribute data, multiple rounds of data preprocessing are executed, including manual searches to fill in missing values and the elimination of duplicate entries. Ultimately, a comprehensive dataset comprising 1836 car models and 22 attributes (*e.g.*, fuel economy, brand, origin, base curb weight, etc.) is compiled. Among these, 1720 are conventional cars, while 119 belong to the category of new energy vehicles (EVs, Plug-in EVs, and hydrogen-powered vehicles).

5.3.2 Step 2: Twitter Data Collection

In this step, we collected data from Twitter based on a reference-based keyword search strategy. The third-party tool, snscrape, in conjunction with Twitter internal query search function,n was utilized to collect tweets from 2016 through 2019. Car models from the reference list (the 949 car model names obtained in Step 1) were the objects searched for in Twitter's database. To allow for consistent samplings across different time periods, a limit of 20 tweets was collected monthly for each car model. This summed up to 240 tweets per car model, with up to 227,760 in total per year. Due to the lack of tweets on specific car models in some months, the number of tweets was less than the maximum number of tweets that were possibly collected. From

2016 to 2019, the number of tweets collected was 86,962; 90,670; 93,861; and 94,302; respectively. The number of tweets increased over the years, influencing the size of the networks generated during this case study.

5.3.3 Step 3: Twitter Data Preprocessing

The pipeline used for Twitter text data preprocessing in this study is shown in Figure 5.6. The first step was removing the URLs from the data frames. This was performed at the outset to simplify the subsequent removal of punctuations. If URLs remained in the data frames, they would be fragmented by the punctuation removal step, rendering their deletion challenging. Then, all punctuation marks were removed. Tokenization was conducted in the third step to split each tweet into individual words. Then, the NLTK library[2] was employed to convert all words to lowercase and remove stopwords, such as "a," "the," and "this". Finally, duplicated tweets were removed from the datasets. These duplicates were deemed to have a high probability of being tweeted by bots whose content was meaningless (Tao et al., 2013). After deleting duplicates, the total number of tweets kept was 34,278; 36,940; 43,347; and 49,895 from 2016 to 2019, respectively.

5.3.4 Step 4: Named Entity Recognition (NER) for Twitter Data

To start this process, we first generated training and testing data using the NER Annotator[3], an online annotation tool, to manually mark the car models in tweets that were processed through Step 3 with the label "CAR." This marking method identifies the beginning and ending indices of each entity in a tweet and subsequently converts this information, along with the tweet content, into the format that is expected by spaCy[4], a mainstream NLP library in Python. We used 2,003 marked tweets from

[2]NLTK: https://www.nltk.org/#
[3]NER Annotator: https://tecoholic.github.io/ner-annotator/
[4]spaCy: https://spacy.io/

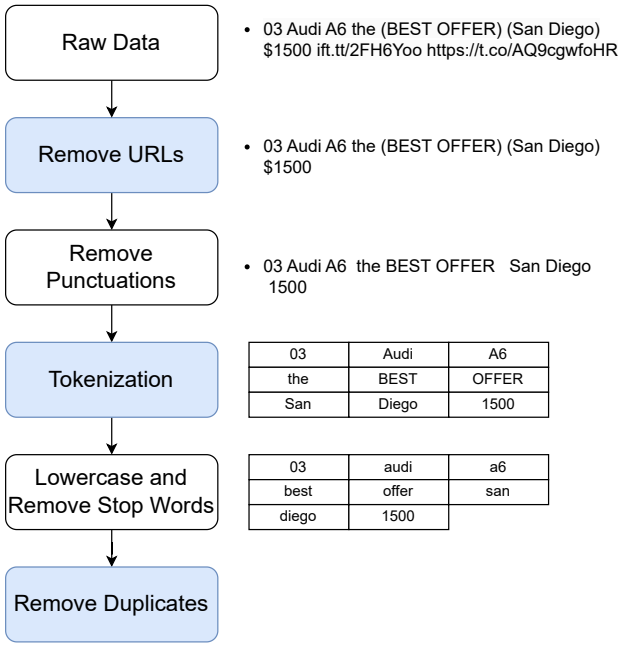


Figure 5.6: Flowchart of the preprocessing method

2018 as training data and 189, 195, 193, and 193 tweets from 2016 to 2019 as testing data, respectively.

The NER model was then trained using the annotated training data. Upon completion of the training phase, model performance was evaluated using independent test data sets spanning 2016 to 2019. The results of these evaluations were tabulated in Table 5.3. The results demonstrate that the model achieved F1-scores greater than 69% over the four years, despite being trained solely on data from 2018. Additionally, all precision values were found to be higher than 74%, indicating that more than 74% of the car models recognized by the NER model were correct identification. Furthermore, all recall values exceeded 66%, signifying that the NER model successfully extracted more than 66% of the ground-truth car models (all the manually marked car models within the testing data) from tweets. Finally, we refer to the results of the Twitter Named Entity Recognition shared task associated with the second Workshop on Noisy User-generated Text (W-NUT 2016) to gain general insights into the overall performance of our model. Their results were generated by ten teams. The average

Table 5.3: The testing results of NER model by year

Year	F1-Score	Precision	Recall
2016	73.25%	80.42%	67.26%
2017	71.50%	77.67%	66.23%
2018	74.83%	74.03%	75.67%
2019	69.96%	74.37%	66.04%

F1-Score of these ten NER models was 38.19%, and the highest value was 52.41% (Strauss et al., 2016). Our model achieved an F1-Score that is approximately 20% higher than the highest F1-Score of the Twitter Named Entity Recognition shared task, justifying the reliability of our trained model.

5.3.5 Step 5: Twitter Co-Mention Network Modeling

In the process of co-mention network modeling, the cleaned four-year tweet data were processed through the trained NER model to identify car model names in each tweet. Subsequently, only tweets containing more than one car model were retained. The resulting count of retained tweets was 4,747; 6,040; 8,408; 11,220 for the years 2016 to 2019, indicating that the percentages of tweets collected that co-mentioned at least two car models were 13.85%, 16.35%, 19.40%, and 22.49%, which is below 50%. This suggests that the co-mention information of cars is dispersed throughout Twitter. Next, given that there existed multiple variant names for some car models in the extracted model name sets, *e.g.*, Ford F 150 being called "Ford F150", "F 150 Ford", and "FordF150", etc., we only generate co-mention connections between identified car models with names consistent with our reference list from Cars.com, resulting in partial entity information loss. For example, we identified three models from one tweet, including "F150 Ford", "Toyota Highlander", and "Subaru Crosstrek". But given that "F150 Ford" differed from the name "Ford F 150" that was recorded in our reference list, we thereby only generated a co-mention connection between Toyota

Highlander and Subaru Crosstrek based on this tweet. [5] Our decision to prioritize an accurate network model over one with more information but greater noise reflects a trade-off. In future work, our aim is to develop a more robust similarity algorithm to address this limitation.

Figure 5.7 illustrates a co-mention network model based on three annotated tweets. The nodes represent unique car models, links signify two car models being co-mentioned in at least one tweet, and link weights denote the total number of tweets that co-mentioned any two models. Note that no sentiment analysis is conducted in this study; thus, the possible relationship between these comorbid car models could be four possible relationships: 1) *association*, denoting that two entities are connected via shared attributes. For example, *My Heart Will Go On* is the theme song for the movie *Titanic*. 2) *Causation*. An example is that the long-term inhalation of specific chemicals is a cause of certain cancers. 3) *Comparison*. For example, two products are co-considered and compared by customers. 4) *Random co-occurrence*, which captures all other undefined relationships. As an example, dealers often announce the arrival of new car models, such as the Toyota Camry, Honda Accord, and Mazda CX-3, and their in-stock status.

5.4 Conclusion

In conclusion, this chapter has addressed the persistent challenge of data scarcity in the engineering and design of socio-technical systems (STSs) by proposing innovative approaches for data collection. By leveraging information retrieval and survey design methods, we have demonstrated two distinct methodologies for gathering networked STS data.

Firstly, we introduced a framework for collecting product co-consideration net-

[5] We utilized text cosine similarity algorithm (Rahutomo et al., 2012) to detect car models and their variants. However, this algorithm is not robust against some models, such as Nissan Z which was difficult to distinguish from other Nissan models due to its name in the short letter "Z".

- Tweet 1: "lexus lc 500[CAR] save get porsche 911 gts[CAR]"
- Tweet 2: "say goodbye my old toyota rav4[CAR] thinking buy new ford f150[CAR] chevy silverado 1500[CAR]"
- Tweet 3: "my friends have toyota rav4[CAR] lexus lc 500[CAR] want subaru wrx[CAR] bad raelene first need learn drive"

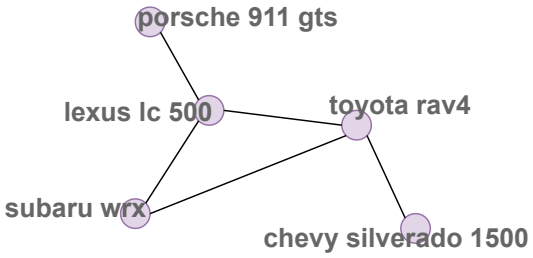


Figure 5.7: An example of co-mention network modeling. These tweets displayed here have undergone processing following Step 3. Nodes are unique car models that were mentioned by all three tweets, and links denote co-mention relationships. For example, a link is built between Lexus lc 500 and Porsche 911 gts because they were co-mentioned by Tweet 1. We did not include "ford f150" in the network modeling because it is inconsistent with the recalled name listed as "ford f 150" in the reference list.

work data, exemplified by the case study of the US household vacuum cleaner market system. This approach, combining web-crawling techniques and survey questionnaire design, facilitated the acquisition of valuable data through a tailored survey web platform. The study yields two primary outputs: 1) a comprehensive vacuum cleaner attribute dataset comprising 624 distinct vacuum cleaner models and 22 associated attributes; and 2) a survey dataset comprising responses from 1002 vacuum cleaner buyers. These responses encompass a wide array of information, including respondents' considerations and purchase decision-making processes, demographics, social network information, and context regarding vacuum cleaner usage.

Secondly, we presented an alternative approach for collecting product co-mentioning network data based on social media mining, focusing on the US vehicle

market system. Utilizing a combination of web-crawling methods and named entity recognition (NER) models, we extracted pertinent network data from textual content sourced from social media platforms. The study yields two main outputs: 1) a comprehensive vehicle attribute dataset featuring 1836 vehicle models spanning from 2016 to 2022. Each car model is described by 22 attributes; and 2) vehicle co-mentioning network data extracted from Tweets, spanning from 2016 to 2019.

Overall, this chapter lays the groundwork for data-driven research in the engineering and design of STSs. The collected data will serve as a foundational resource for subsequent chapters, enabling further exploration and analysis of socio-technical systems dynamics.

Chapter 6: Micro-Level Entity Design Considering Meso-Level Dependencies

6.1 Overview

Unlike existing efforts that have primarily focused on the network-based analysis of STSs (*i.e.*, the forward problem). The objective of this chapter is to solve the inverse problem, *i.e.*, how can we achieve the desired system-level performance by promoting the formation of targeted relations among local entities? To achieve this goal, we developed a network-based STS design framework. This framework uses network representations and network motif theory to characterize and capture local dependencies and relations between individual entities in STSs and integrate these representations into design formulations to find optimal decisions for the desired functionality of individual entities. The development of this framework contributes to answering ***RQ1***, ***RQ2***, and ***RQ3***.

To demonstrate its utility, we applied this framework to the design of market systems with a case study on vacuum cleaners. The objective was to promote the frequency of product purchases by optimizing suction power and weight while maintaining constant prices. Specifically, we innovatively proposed a derived design parameter that incorporates local competition information into the design process. We addressed this problem by integrating an exponential random graph model (ERGM) with a genetic algorithm. Compared to traditional design methods that do not consider local competition relationships, the results indicate that the new designs can effectively increase the frequency of product purchases.

In detailed, this chapter is organized as follows:

- Section 6.2 introduces the proposed micro-level entity design framework considering meso-level dependencies step-by-step.

- Section 6.3 adopts the US household vacuum cleaner market system as a case study to demonstrate the design framework.
- Section 6.4 discusses the limitations of the proposed framework and clarifies its constraints in supporting system design.
- Section 6.5 concludes this chapter with closing thoughts.

6.2 Network-Based System Design Framework

In Figure 6.1, we compare a typical system design process with our proposed network-based approach. The traditional method begins by analyzing the system requirements from which the design goal is set and the design variables and constraints are identified (Arora, 2004; Martins and Ning, 2021). The design goal guides the formulation of the design objective function and the design variables and constraints define the design space. Subsequently, optimal/suboptimal design solutions are found by exploring and exploiting the design space by evaluating design candidates with the objective function. Finally, the design solution is validated against system-level requirements. Building upon the traditional framework, the proposed network-based design framework consists of six major steps, each of which is elaborated below.

6.2.1 Step 1: Network modeling and system design goal definition

The primary objective of the first step is to create a network representation, labeled as $\boldsymbol{Y}(\boldsymbol{X})$, in which $\boldsymbol{Y}$ corresponds to the network's adjacency matrix, $\boldsymbol{X}$ represents the vector of system design attributes. $\boldsymbol{Y}$ changes with $\boldsymbol{X}$. Let us take the customer-product market system as an example, in which the co-consideration relations among products can be represented by a unidimensional network $\boldsymbol{Y}$ shown in Figure 6.2. This network is built using data from customers' considerations of vacuum cleaners and following the approach described in (Xiao et al., 2022b). In such a network, each node represents a unique product model that the customers consider. The

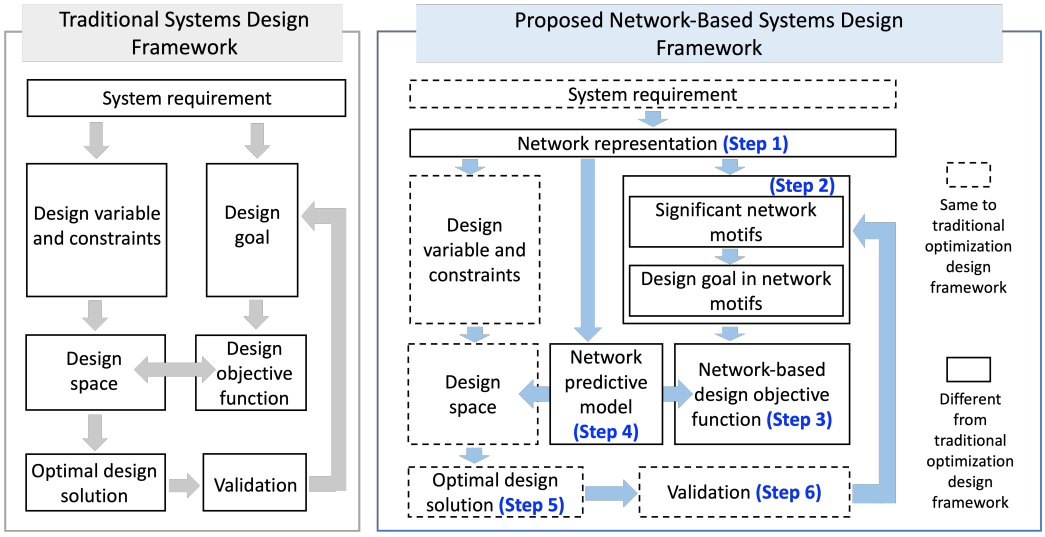


Figure 6.1: Comparison between the traditional system design framework and the proposed network-based system design framework with considering local dependencies

dashed links denote that two products from different brands (*e.g.*, Dyson vs. iRobot) are co-considered by at least one customer. In contrast, the solid links denote co-consideration within the same brand. We assume that the design attribute vector $\boldsymbol{X} = [x_1, x_2]$ in this example only includes the suction power and weight of each product. Updating the design attribute vector $\boldsymbol{X}$ for any product will influence the co-consideration and therefore the network structure $\boldsymbol{Y}$.

6.2.2 Step 2: Representing the design goal using network motifs

The objective of Step 2 is to represent the design goal using local networks. This involves transforming the original design objective function $u(\boldsymbol{X})$ into a function of local networks, denoted as $u(\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})))$, where $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X}))$ indicates the derived local network-based design variable (either a scalar or a vector). This step is essential to incorporate the significant dependencies between individual entities (represented by local networks) into the design process. To achieve the objective, we first identify significant local networks $\boldsymbol{y}(\boldsymbol{X})$ based on network motif theory, which is introduced in Section 2.3.1.

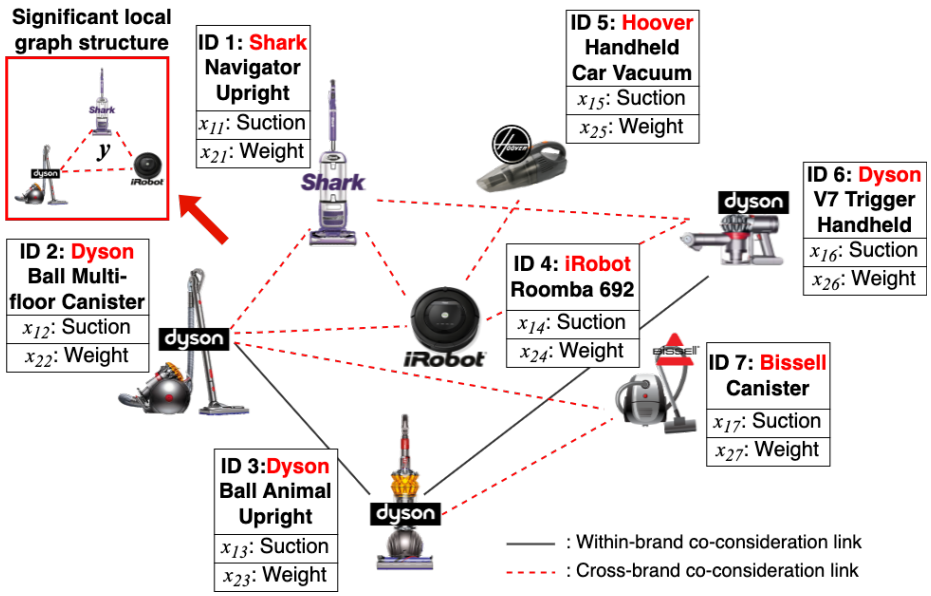


Figure 6.2: An example of the customer-household vacuum cleaner market co-consideration network.

Having identified the significant motifs, we proceed to convert the original design goal $u(\boldsymbol{X})$ into the new representation in the form of local networks $u(\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})))$, which capture interdependencies among individual entities (*e.g.*, product models). To illustrate this, let us use an example of the vacuum cleaner market system. Suppose that we take on the role of Dyson, with the original design goal $u(\boldsymbol{X})$ being to participate in the dominant product competition as much as possible. If a significant network motif $\boldsymbol{y}$ (shown in the top-left corner of Figure 6.2), representing the inter-brand triadic closure competition, is identified as the significant competition pattern in this vacuum cleaner market, then we transform our goal into maximizing the number of inter-brand triadic closure competitions a product involving in. Next, assume that the number is negatively correlated with the average suction power difference between products within the triadic closure. For a target product (*e.g.*Product 2), the local network-based design variable $\boldsymbol{g}$ becomes a scalar $g(\boldsymbol{y}(\boldsymbol{X})) = \frac{1}{3}[|x_{12} - x_{11}| + |x_{12} - x_{14}| + |x_{11} - x_{14}|]$ and $\boldsymbol{X} = [x_{12}]$ (the values of x_{11} and x_{14} are given). The transformed design objective function represented by local networks is thus $u(g(\boldsymbol{y}(\boldsymbol{X}))) = -\frac{1}{M(\boldsymbol{X})}\sum_{m=1}^{M(\boldsymbol{X})} g_m(\boldsymbol{y}(\boldsymbol{X}))$. The negative sign

indicates the assumed negative correlation between u and g. M is the total number of inter-brand triadic closure competitions in which Product 2 is involved.

6.2.3 Step 3: Optimization problem formulation

In Step 3, an optimization problem is formulated based on the local network-based design objective function obtained in Step 2. Figure 6.3 illustrates an example of the optimization problem associated with the transformed design objective $u(g(\boldsymbol{y}(\boldsymbol{X})))$. The objective function $f(\boldsymbol{X})$ is defined to maximize the number of participations of all Dyson products in inter-brand triadic competitions, u_i, by adjusting the design attributes, x_{1i} and x_{2i}, for each product. $i = 2, 3, 6$ stands for the product IDs of all Dyson products in Figure 6.2. However, solving this network-transformed optimization problem is a challenge. This is because to obtain the number of triadic competitions $M(\boldsymbol{X}_i)$ in which the product i participates, we must know the network topology. But it changes every time when we change the design attributes (*e.g.*, suction power stored as a node feature), and there is a lack of analytical expression between the design attributes and the network structure. Therefore, solving such an optimization problem that contains design variables in network representations (that could be non-linear) necessitates the employment of a surrogate model to predict new network structures when a node feature changes.

6.2.4 Step 4: Predictive model training and evaluation

Step 4 is to establish a model to predict the new network topology after updating the design attributes $\boldsymbol{X}$. Typical network models, such as ERGM (Section 2.3.2) and graph neural networks (GNN)(Section 2.3.3), can be used for this purpose. In this study, we have chosen to adopt ERGM due to its verified performance in our previous work (Sha et al., 2023), we will explore other models in our future work.

As introduced in Section 2.3.2, in an ERGM, $\boldsymbol{g}(\boldsymbol{Y_{obs}})$ is a vector of the network statistics that can encompass either nodal attributes or edge attributes asso-

Optimization Problem Example

$$\max f(\boldsymbol{X}) = \max[u_2\left(g\left(\boldsymbol{y}(\boldsymbol{X}_2)\right)\right) + u_3\left(g\left(\boldsymbol{y}(\boldsymbol{X}_3)\right)\right) + u_6\left(g\left(\boldsymbol{y}(\boldsymbol{X}_6)\right)\right)]$$

S.t. $\boldsymbol{X}_i = [x_{1i}, x_{2i}], i = 2,3,6$

$$x_{suc\_low} \leq x_{1i} \leq x_{suc\_high}$$

$$x_{weig\_low} \leq x_{2i} \leq x_{weig\_high}$$

$u_i = -\frac{1}{M(\boldsymbol{X}_i)}\sum_{m=1}^{M(\boldsymbol{X}_i)} g_m\left(\boldsymbol{y}(\boldsymbol{X}_i)\right)$: the number of inter-brand triadic closure competitions that product i participates in, represented by the average suction difference among products within the triadic competitions. Note that the co-consideration network topology changes with product attributes. The number of triadic competitions $M(\boldsymbol{X}_i)$ that product i engaged in is therefore the function of the attribute vector $\boldsymbol{X}_i$.

Figure 6.3: An example of optimization problem formulation with local network-based design objective

ciated with design attributes $\boldsymbol{X}$. However, it should be noted that the successful construction of the target ERGM is based on two factors. First, it is determined by the design attributes $\boldsymbol{X}$. This means that the attributes intended for the design must be included in the model through network statistics, such as nodal attribute effects introduced in Table 2.3. Second, achieving a converged ERGM with satisfactory predictive power requires a trial-and-error process[1]. Similar to Table 2.3, three types of network statistic examples and associated interpretations in the context of

[1]An ERGM failing to converge indicates that the parameter estimates are not settling down to stable values, and the iterative estimation process is not reaching a consistent solution. Typical reasons for the convergence issue of ERGM include model degeneracy (an ill-fitting model in ERGM fails to adequately represent the observed network) (Hunter, 2007) and inappropriate model specifications (Butts et al., 2014).

the vacuum cleaner co-consideration network are given in Table 6.1.

Table 6.1: Three major network statistics in ERGM with their examples (Sha et al., 2023; Morris et al., 2008)

Category	**Examples**	**Interpretation**
Nodal attributes effects	*Nodecov*	Main effect of a covariate. For example, *nodecov.suction* understands how suction power influences a vacuum cleaner being co-considered with other vacuum cleaners.
Relational attributes effects	*Absdiff*	Absolute difference between two connected nodes' attributes. For example, *absdiff.weight* looks into whether a large or small weight difference between two vacuum cleaners motivates them to be co-considered.
Network structural effects	*Edges*	Equal to the number of links in the network, equivalent to the intercept term in the regression model. In the context of the vacuum cleaner co-consideration network, it estimates the likelihood that two vacuum cleaners will be co-considered randomly.
	GWESP	Geometrically weighted edgewise shared partner. In (Hunter, 2007), it is also called k-triangle, which is defined to be a set of k distinct triangles that share a common edge. The *GWESP* term models the tendency for edges that close triangles to be more probable than edges that do not close triangles. In the context of the vacuum cleaner co-consideration network, it investigates whether two vacuum cleaners co-considered with the same set of vacuum cleaners are more likely to be co-considered or not.

ERGM training involves estimating the model parameters $\boldsymbol{\theta}$ by feeding the

network data obtained from Step 1. Once the estimated parameters are obtained, the predictive performance of the estimated model can be evaluated in four steps. The first step is to use the estimated model to simulate N number of networks ($\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$,$\cdots$, $\boldsymbol{Y}_N$). Then, in the second step, each simulated network $\boldsymbol{Y}_n$ ($n = 1, \cdots, N$) is compared against the observed network $\boldsymbol{Y}_{obs}$ (that is, the ground truth) in Step 1 to classify all possible links of the network in a confusion matrix (Goodfellow et al., 2016). According to the confusion matrix, we can then calculate common metrics, such as *Recall, Precision*, and *F1-Score*, to evaluate the predictive performance of the model (Goodfellow et al., 2016). Finally, the mean values of *Recalls, Precisions*, and *F1-Scores* of those N simulated networks are used to represent the performance of a trained ERGM.

6.2.5 Step 5 and Step 6: Optimal problem solving and solution validation

In Step 5, the model obtained from Step 4 will be used as a surrogate model to predict new network structures in order to re-evaluate the objective value $f(\boldsymbol{X})$ after modifying the associated node features (*i.e.*, design attributes). Consequently, the computational search process for the optimal solution is performed through meta-heuristic approaches, such as the genetic algorithm (Whitley, 1994) or particle swarm optimization (Kennedy and Eberhart, 1995). After finding the optimal design solution in Step 6, it can be validated by implementing the new designs and observing if the desired system performance can be achieved or not. For example, in the vacuum cleaner example, we will count if the new design attributes of a particular product would help increase its participation in the desired competition relations on the market. Validation is often challenging as it requires real-world implementation and testing. In this paper, we focus primarily on the verification of the optimization results computationally by recalculating the design objective with the original and optimal design variables and checking if the objective value of the optimal design indeed increases or not.

6.3 Case Study

In this section, we utilize the data collected on the US household vacuum cleaner market system, as detailed in Section 5.2, to illustrate the proposed network-based system design framework.

6.3.1 Network Modeling

Co-consideration network model In this study, we are interested in competition analysis in the vacuum cleaner market system. We, therefore, construct the co-consideration network following our previous study (Xiao et al., 2023b). In this unidimensional network, the nodes are unique vacuum cleaners from the top ten dominant brands and are considered by customers. Similar to the example given in Figure 6.2, the undirected links represent that two vacuum cleaners are co-considered by at least one customer. The visualization of the co-consideration network is shown in Figure 6.4. This network contains 386 unique vacuum cleaner models and 1259 co-consideration links. Product 369, Dyson Ball Multi-floor 2, has the largest degree, indicating that it is co-considered most frequently.

6.3.2 Deriving the local network-based design goal and formulating the optimization design problem

Definition of derived local network-based design variable As described in Section 6.2.2, the first step of deriving the local network-based design goal is to identify significant local network structures (Xiao et al., 2023b). As shown in Table 6.2, three significant network motifs of the co-consideration network are identified by the motif mining tool, FANMOD (Wernicke and Rasche, 2006), each of which represents distinct competition relationships between brands (inter-brand) and within a brand (intra-brand). They are named for their edge types and topological characteristics. A real-world example for each significant motif is also given in Figure 6.4. Among the three motifs, the inter-brand triadic closure competition with the highest *Z-score*

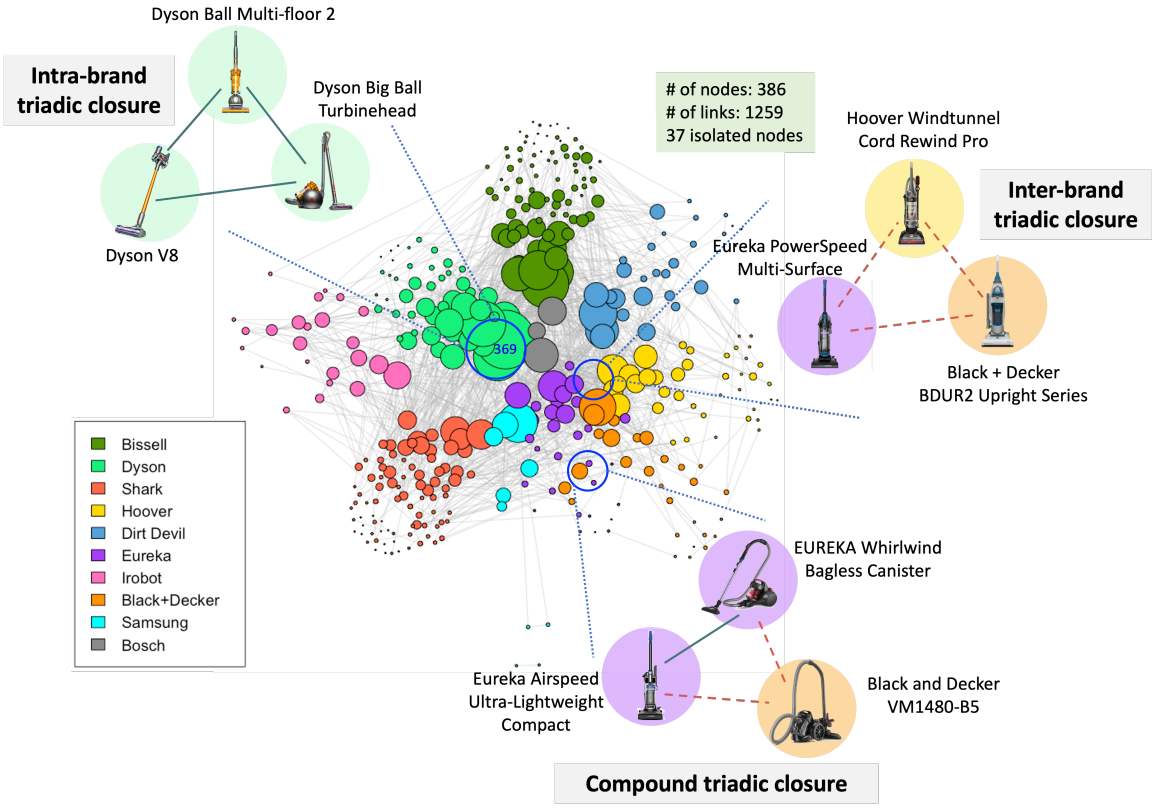
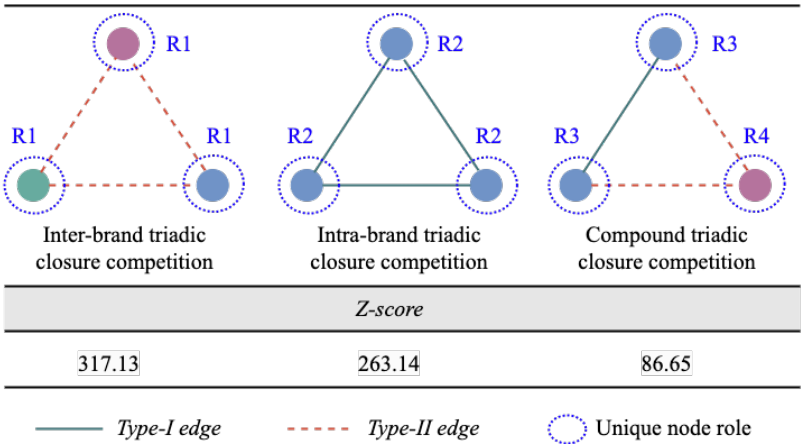


Figure 6.4: Co-consideration network of top-ten household vacuum cleaner brands is found to be the most significant competition structure in the co-consideration network. Based on the positional characteristics in these significant motifs, we can define four unique node roles: $R1$, $R2$, $R3$, and $R4$. For example, in the inter-brand triadic closure, all three node positions share the same type of node role, $R1$, because each node is co-considered with two products from two other brands in a closed triangle competition. Accordingly, we define the derived local network-based design variable of each product as $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}, N_{R2}, N_{R3}, N_{R4}]$, where N_{Ri} (for $i = 1, 2, 3, 4$) is the number of times a product is involved in the node role Ri. This defined network-based design variable can be easily extended. For example, if additional node roles such as $R5$ and $R6$ are discovered, we can extend the derived variable by concatenating N_{R5} and N_{R6}, resulting in $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}, N_{R2}, N_{R3}, N_{R4}, N_{R5}, N_{R6}]$. Similarly, the vector can be shortened by omitting less important node roles.

Table 6.2: Significant size-3 competition motifs in the co-consideration network and the unique node roles inherent in the motif structures. The *type-I edge* indicates that two vacuum cleaners share the same brand, and *type-II edge* refers to the different brands.



Optimization problem formulation Now, let us pick one particular product model, *e.g.*, Dyson Ball Multi-floor 2 (Product 369) – the one with the most co-consideration connections in the observed network, to continue the demonstration due to its increasing popularity in the US market. Assuming that Dyson is interested in maximizing a product's market share, we use the number of purchases as an indicator of that product's market share. Next, we formulate the network-based design objective function by estimating the relationships between the number of times product purchases u and the local network-based design variable derived $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X}))$. As shown in Table 6.2, given that inter-brand triadic closure shows the highest *Z-score*, we simplify the derived design variable by focusing only on the node role $R1$ in our first test case, resulting in $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}]$. In this study, since the data format of the number of times product purchases is a count, following the method introduced in (Elhai et al., 2008), negative binomial regression is selected to estimate the relationship between u and $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X}))$. To ensure the reliability of the estimate model, four models corresponding to the combination of polynomial terms of the independent variable up to cubic are tested. Both the mean absolute error (MAE) (Chai and Draxler, 2014) and Akaike's Information Criterion (AIC) (Bozdogan, 1987) are ap-

plied to measure the goodness of fit of the model while considering a balance between the goodness of fit and model complexity. The model with the lowest AIC and MAE is finally selected, which is provided in Table 6.3.

Table 6.3: Negative binomial regression estimated result of u corresponding to $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}]$.

Independent Variables	**Est. Coef.**	**Std. Error**
Intercept	0.316***	0.075
N_{R1}	0.117***	0.017
N_{R1}^2	-0.002***	0.0005

***: 0.000 level of significance

Therefore, the design objective function for the derived local network-based design variable is given in Equation (6.1).

$$u(\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X}^{P369}))) = exp(0.316 + 0.117N_{R1}^{P369} - 0.002(N_{R1}^{P369})^2), \tag{6.1}$$

where N_{R1}^{P369} is the number of times Product 369 is involved in node role $R1$. This is equivalent to the number of inter-brand triadic closure competitions that include Product 369. Thus, we can further express N_{R1}^{P369} as shown in Equation (6.2).

$$N_{R1}^{P369}(\boldsymbol{y}(\boldsymbol{X}^{P369})) = \frac{1}{2}\sum_{m=1}^{M} det(\boldsymbol{y}_{inter\_brand}^{m}(\boldsymbol{X}^{P369})). \tag{6.2}$$

In this equation, M represents the number of all potential inter-brand triadic closure competitions in which Product 369 is involved. These competitions can be enumerated when we know the total number of products on the market. Suppose that there are K products on the market, and we denote the product set as V where each element is a product model named by its ID (*e.g.*, product 369's ID name is $P369$). In that case, we can pre-save the inter-brand triadic closure competitions in which $P369$ is involved as a set denoted as $S = \{\boldsymbol{y}_{P369,V_i,V_j}^{m}\}$ where V_i and V_j represents the i^{th} and j^{th} products in V with which Product 369 competes. For instance, in

the example shown in Figure 6.2, V denotes $\{P1, P2, P3, P4, P5, P6, P7\}$, so $K = 7$. Among these products, P1, P4, P5, and P7 are the only products that can form the inter-brand triadic closure competition with Product 2. As a result, $M = 6$ for Product 2, and $S = \{\boldsymbol{y}^1_{P2,P1,P4}, \boldsymbol{y}^2_{P2,P1,P5}, \boldsymbol{y}^3_{P2,P1,P7}, \boldsymbol{y}^4_{P2,P4,P5}, \boldsymbol{y}^5_{P2,P4,P7}, \boldsymbol{y}^6_{P2,P5,P7}\}$.

In the case study, according to the network model shown in Figure 6.4, we get $K = 385$ and $M = 46,650$ for Product 369. $\boldsymbol{y}_{inter\_brand}$ represents the adjacency matrix for the inter-brand triadic closure competitions. It equals to $\left(\begin{smallmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{smallmatrix}\right)$, when the triadic closure exists. Accordingly, the determinant of $\boldsymbol{y}_{inter\_brand}(\boldsymbol{X}^{P369})$, denoted as $det(\boldsymbol{y}_{inter\_brand}(\boldsymbol{X}^{P369}))$, equals 2. The existence of the inter-brand triadic closure is determined by the original product design vector $\boldsymbol{X}^{P369}$. In this case study, we focus on two essential design attributes of vacuum cleaners: suction power (x_s) and weight (x_w). Therefore, $\boldsymbol{X}^{P369} = [x_s^{P369}, x_w^{P369}]$. In addition, we factor in one constrained variable: price (x_p). The reason for having this constraint is that price is one of the most important factors influencing customer choice behaviors, and keeping it unchanged helps avoid the price perception bias that customers have toward the new design (Kotler et al., 2014). Finally, the formulated optimization problem is presented in Figure 6.5.

6.3.3 ERGM-based network prediction

As aforementioned, this study focuses on two design attributes and one constraint: suction power level (x_s), weight (x_w), and price (x_p). Therefore, they are taken into account by incorporating their associated *Nodecov* and *Absdiff* terms (Robins et al., 2007), as introduced in Table 6.1. To ensure model convergence, a trial-and-error process is performed. We evaluated 27 models with varying combinations of *Nodecov* and *Absdiff*, ultimately pinpointing a converged ERGM with all terms achieving a level of significance (*p-value*) close to 0. As presented in Table 6.4, in addition to three nodal effect terms, two network effect terms given in Table 6.1, *Edges* and the *GWESP*, are also included. To facilitate the convergence of the model and

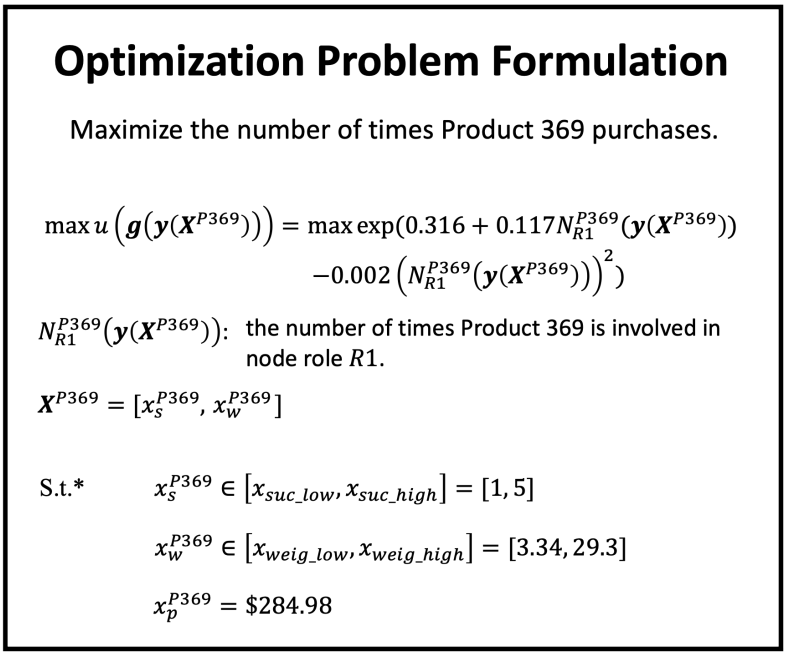


Figure 6.5: The optimization problem formulation.

improve its performance, max-min normalization is applied to preprocess the attribute data (Dutka and Hansen, 1991). The estimated results, *i.e.*, the estimated model parameters $\boldsymbol{\theta}$ in Equation 2.3, are shown in Table 6.4. For example, the negative sign of *Absdiff.price* shows that two vacuum cleaners with less difference in their prices are more likely to be co-considered. In contrast, the positive sign of *GWESP* means that two vacuum cleaners that share the same set of co-consideration products are more likely to be co-considered with each other. It implies that customer consideration decisions involve a form of multiway grouping and comparison (Sha et al., 2018).

Once a trained model is obtained, we follow the process introduced in Section 6.2.4 to simulate 100 networks and validate its predictive power. By comparing the simulated networks against the ground truth, we calculate the means of the *Precisions*, *Recall*, and *F1-Scores* of all 100 simulated networks, and the results are 0.158, 0.192, and 0.173 respectively. The *Precision* 0.158 indicates that 15.8% predicted co-consideration links are correctly predicted on average. The *Recall* 0.192 means that

Table 6.4: Estimated result of the predictive ERGM

Independent Variables	**Est. Coef.**	**Std. Error**
Edges/Intercept	-6.717***	0.105
Absdiff.price	-0.562***	0.148
Absdiff.weight	-1.287***	0.194
Nodecov.suction	0.187***	0.035
GWESP	2.671***	0.092

***: 0.000 level of significance

19.2% truly existing links are correctly predicted. Lastly, *F1-Score*, the harmonic mean of *Precision* and *Recall*, evaluates a balanced predictive accuracy of the model. It should be noted that in this study we did not spend excessive resources to find the best ERGM model and only included four independent variables in the model, with the objective of this case study being to demonstrate the proposed design framework. Therefore, we focused on more model convergence and stopped testing additional models (which required more data collection efforts) for better prediction. A further discussion of this is presented in detail in Section 6.4.

6.3.4 Optimal design solutions

After having a predictive model, the next step in solving the optimization problem involves calculating the objective value u each time the design variables (*i.e.*, the values of suction power and weight of Product 369) are explored. As illustrated in Algorithm 1, we first employ the trained ERGM model with the estimated model parameters $\boldsymbol{\theta}_{est}$ provided in Table 6.4 to simulate 100 networks $\boldsymbol{Y}_l$. Within this set of networks, we examine each triadic closure $\boldsymbol{y}^m_{P369,V_i,V_j}$ contained in the set S. Our objective is to count its occurrence across the 100 networks and compute its occurring ratio, representing its probability of existence, denoted as $Pr(\boldsymbol{y}^m_{P369,V_i,V_j})$. Given that $Pr(\boldsymbol{y}^m_{P369,V_i,V_j})$ typically exhibits a skewed distribution with most values low, we set the median of these probabilities as the threshold to determine the existence of each triadic closure. This choice ensures a more accurate measure of central tendency and provides robustness to outliers (Von Hippel, 2005). Accordingly, the number

of existing inter-brand triadic closures, equivalent to N_{R1}^{P369}, is obtained and incorporated into Equation (6.1) to compute the objective value. Next, as described in Algorithm 2, the evaluation of the objective function is built into the genetic algorithm (GA) (Whitley, 1994) which helps to find the optimal level of suction power and weight with the constraint on price to maximize the objective value. The initialization of the GA algorithm is detailed in Algorithm 2, where '*popSize*' denotes the population size in each round of search, '*maxiter*' indicates the defined maximum number of generations to run before the GA search stops, and '*run*' means the maximum number of consecutive generations for which the best objective value (fitness) has no improvement, leading to the termination of the GA search (Whitley, 1994).

Algorithm 1 Objective Value Calculation

Given V, S, x_s^{P369}, x_w^{P369}, x_p^{P369}, $\boldsymbol{\theta}_{est}$
Initiate $L = 100$
Simulate L networks $\boldsymbol{Y}_l, (l = 1, ..., L)$ with the given x_s^{P369}, x_w^{P369}, x_p^{P369}, and estimated ERGM parameters $\boldsymbol{\theta}_{est}$
for $m = 1$ to M **do**
    $count = 0$
    for $l = 1$ to L **do**
        if $\boldsymbol{y}^m_{P369,V_i,V_j}$ exists in $\boldsymbol{Y}_l$ **then**
            $count = count + 1$
        end if
    end for
    $Pr(\boldsymbol{y}^m_{P369,V_i,V_j}) = count/L$
end for
$Pr_{threshold} = \text{Median}(Pr(\boldsymbol{y}^m_{P369,V_i,V_j}))$
for $m = 1$ to M **do**
    $N_{R1}^{P369} = 0$
    if $Pr(\boldsymbol{y}^m_{P369,V_i,V_j}) > Pr_{threshold}$ **then**
        $N_{R1}^{P369} = N_{R1}^{P369} + 1$
    end if
end for
Return $u = exp(0.316 + 0.117N_{R1}^{P369} - 0.002(N_{R1}^{P369})^2)$

Algorithm 2 Optimization Process

Constraint x_p^{P369}
Variable $\boldsymbol{X} = [x_s^{P369}, x_w^{P369}]$
fittness = function($\boldsymbol{X}$) + Objective Value Calculation($\boldsymbol{X}$)
GA (type = "real-valued",
    fittness,
    min = $[x_{suc\_low}, x_{weig\_low}]$,
    max = $[x_{suc\_high}, x_{weig\_high}]$,
    popSize = 30, maxiter = 100, run = 15)
Summary (GA)

6.3.5 Comparison between the traditional and proposed design methods

In this section, we compare and evaluate the design outcomes between the traditional and proposed design methods. The key difference between these methods is that the traditional approach optimizes product design by treating each product independently, relying solely on the relationship between the design objective (*e.g.*, maximizing market share) and product attributes. In contrast, the proposed method also considers local dependencies between products, such as competition, during the optimization process. Two design cases were analyzed using both methodologies. Case One focuses on designing the suction power of Product 369 to enhance its market competitiveness, considering the constraints on weight and price. In Case Two, we optimize both the suction power and the weight of Product 369 with the objective of increasing its likelihood of being purchased, while keeping its price unchanged.

Results of traditional design method Regarding the traditional method, we adhere to the procedure conducted in Section 6.3.2 to directly estimate the relationship between the number of times product purchases u and the original design vector $\boldsymbol{X}$ using a negative binomial regression model, without local network representation of competition relations between products. Specifically, for Case One, $\boldsymbol{X} = [x_s]$, and for Case Two, $\boldsymbol{X} = [x_s, x_w]$. Furthermore, we incorporate the price (x_s) into the model due to its role as the constrained variable. The estimated results for both cases are presented separately in Table 6.5 and Table 6.6.

Table 6.5: Case One: negative binomial regression estimated result of u corresponding to $\boldsymbol{X} = [x_s]$

Independent Variables	**Est. Coef.**	**Std. Error**
Intercept	1.110***	0.310
x_s	-0.432.	0.231
x_s^2	0.087*	0.039
x_p	0.0003	0.0003

***: 0.000 level of signifi. *: 0.01 level of signifi. .: 0.05 level of signifi.

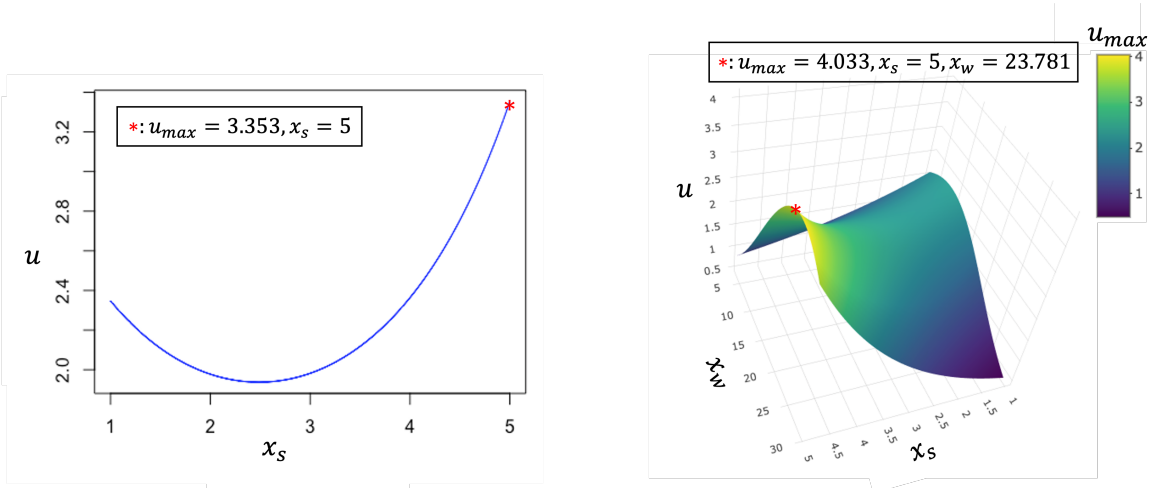
Table 6.6: Case Two: negative binomial regression estimated result of u corresponding to $\boldsymbol{X} = [x_s, x_w]$

Independent Variables	**Est. Coef.**	**Std. Error**
Intercept	0.764**	0.295
x_s	-0.317*	0.135
x_p	0.0004	0.0003
x_w	0.038	0.035
x_w^2	-0.004*	0.002
$x_s x_w$	0.028**	0.011

**: 0.001 level of signifi. *: 0.01 level of signifi.

Next, in Case One, we take the original price and weight of Product 369 (x_p = \$284.98, x_w = 15.6 LB) into the estimated regression model in Table 6.5, aiming to search the maximum u value within the specified suction power design range [1, 5][2]. In Case Two, we maintain the original price constraint while relaxing the weight constraint. Our goal is to identify the maximum u value within the design space defined by the suction power range [1, 5] and the weight range [3.34 LB, 29.3 LB]. According to Figure 6.6 (a), the highest $u = 3.353$ occurs at $x_s = 5$. In Figure 6.6 (b), the highest $u = 4.033$ is achieved when $x_s = 5$ and $x_w = 23.781$ LB.

[2]In the dataset, vacuum cleaners of different brands or categories have different units for suction power. Two commonly used units are horsepower and airflow (bes)). To solve the problem, we first unify the suction powers of the same unit in the range $[1, 5]$ without units. For example, if the original airflow interval is [21.2 CFM, 160 CFM] (CFM: cubic feet per minute), we evenly divide it into five subintervals. Products with airflow in the first subinterval [21.2 CFM, 48.96 CFM] are assigned a level value of 1, and the same operation applies to levels 2, 3, 4, and 5. In cases with multiple suction power units for the same product, the final level value is the average of the level values converted from multiple units.

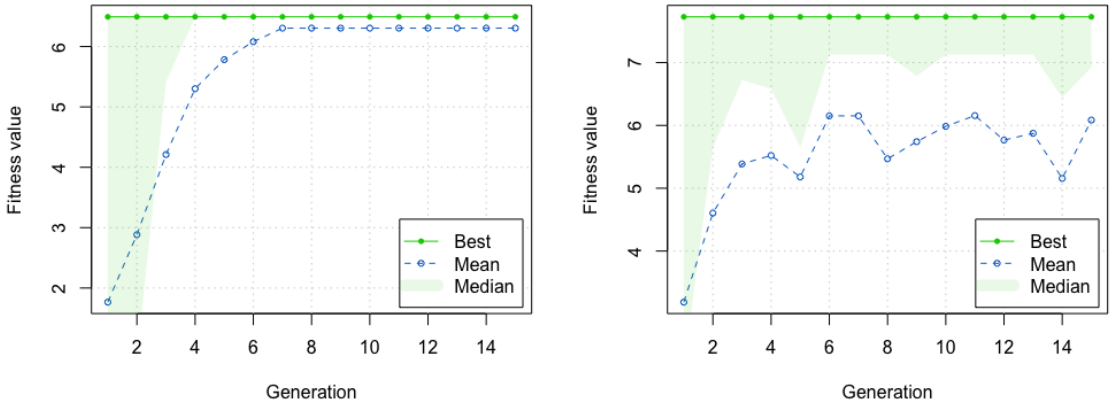


(a) Case One: suction power design case

(b) Case Two: suction power and weight design case

Figure 6.6: Optimal design solutions obtained by traditional design method.

Results of the proposed design method Following the methodology outlined in the preceding subsections, we execute the algorithm detailed in Section 6.3.4 to explore optimal design solutions. Figure 6.7 shows the converged search process for optimal values. In Case One, the search terminates in the 15^{th} generation as there is no improvement in the best objective value for 15 consecutive generations. Throughout these generations, we identify ten optimal suction power values at [2.676, 2.680, 2.684, 2.688, 2.692, 2.696, 2.704, 2.708, 2.712, 2.716] (decrease by 1.284 to 1.324 from its original design), which corresponds to the best objective value of $u = 6.493$. In Case Two, convergence is achieved by the 15^{th} generation as well. We identify three sets of optimal solutions characterized by weight and suction power values $[x_s, x_w]$: [4.291, 22.317 LB], [4.445, 28.457 LB], and [1.985, 20.078 LB]. These solutions align with the best objective value of 7.729. Lastly, according to the mean value curves of the iterative search processes for both cases, we can observe that the search process in Case Two is more fluctuant. This fluctuation could be attributed to the fact that the design space of Case Two, corresponding to two product attributes, is 2D and therefore much larger and more complex than the 1D design space of Case One.



(a) Case one: suction power design case

(b) Case two: suction power and weight design case

Figure 6.7: Iterative search processes using a genetic algorithm to optimize two design cases. The processes terminate after 15 generations, respectively, with a convergence criterion of no improvement in the best objective value (fitness) for 15 consecutive generations. In the plot, the lower boundary of the green shaded area represents the median fitness value, while the upper boundary corresponds to the best (maximum) fitness value. This shaded area delineates the range within which the fitness values of the top 50% of the population fall in each generation. Consequently, it visualizes the spread and variability of the fitness values within the upper half of the population.

Results comparison A comparison of the final results between the traditional and proposed design methods is summarized in Table 6.8 (The columns corresponding to the Traditional Design Method and Proposed Design Method for $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}]$). According to the table, the proposed design method achieves objective values approximately twice as high as those of the traditional design method in both cases. For instance, in Case Two, the proposed design method, considering local inter-brand triadic closure competition relationships, achieves an objective value of 7.729. If translating to the number of times purchases, this is approximately eight times, about two times higher than the traditional design method (4.033). Moreover, the objective values obtained using the proposed method for both cases significantly exceed those derived from applying the original design to the proposed algorithm. This further

demonstrates the efficiency of the proposed design method. Lastly, the optimal design of Case Two using the proposed design method provides three design options, which offer varying trade-offs between suction power and weight, allowing for tailored solutions catering to different customer preferences while maximizing the product's market appeal.

6.3.6 Extensibility of the proposed design method

In this section, we demonstrate the extensibility of the proposed design method by considering the second most important node role, $R2$, as shown in Table 6.2. Consequently, the derived local network-based design variable changes from $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}]$ to $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}, N_{R2}]$. Accordingly, the optimal design problem is reformulated by estimating the relationship between u and the new $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X}))$.

Optimal design reformulation and solution The estimated result is provided in Table 6.7. The new local network-based design objective function for the derived variable is illustrated in Equation (6.3). The number of times Product 369's involvement in node role $R2$, $N_{R2}^{P369}(\boldsymbol{y}(\boldsymbol{X}^{P369}))$, can be expressed similarly to $N_{R1}^{P369}(\boldsymbol{y}(\boldsymbol{X}^{P369}))$ in Equation (6.2). In the network model shown in Figure 6.4, 51 unique products, including Product 369, are from Dyson, resulting in 1,225 potential intra-brand triadic closure competitions for Product 369. Keeping the rest of the settings unchanged, the updated optimization problem is provided in Appendix C, Figure C1. Solving this updated optimization problem follows the same logic as introduced in Section 6.3.4, with minor revisions to Algorithm 1 while keeping Algorithm 2 unchanged. Since most steps are the same, we do not repeat them here and have included the updated Algorithm 1 in Appendix C.

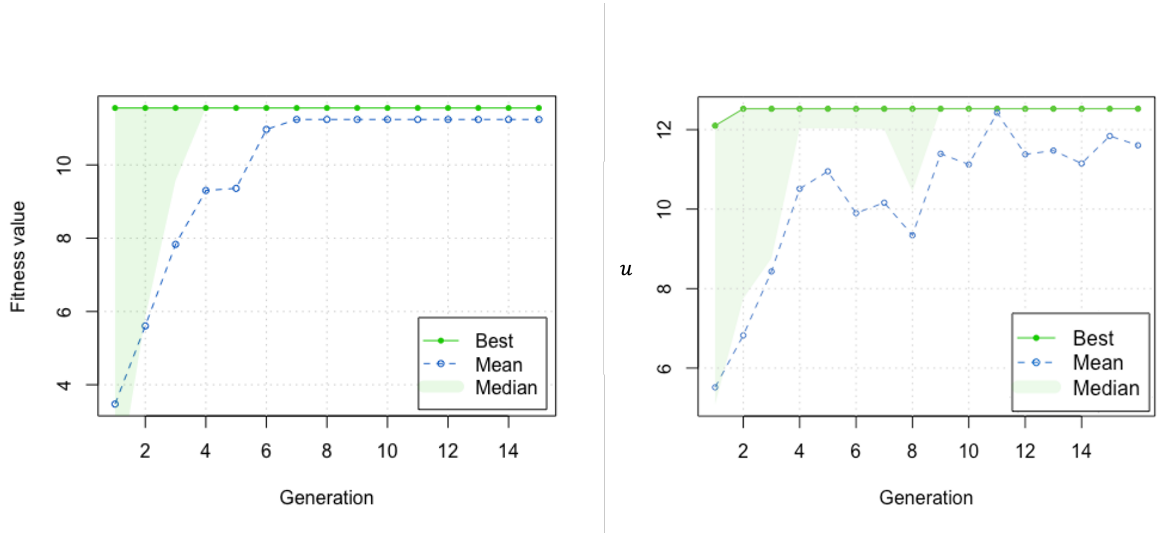
$$u(\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X}^{P369}))) = exp(0.227+0.097N_{R1}^{P369}-0.002(N_{R1}^{P369})^2)+0.204N_{R2}^{P369}-0.014(N_{R2}^{P369})^2). \tag{6.3}$$

Table 6.7: Negative binomial regression estimated result of u corresponding to $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}, N_{R2}]$.

Independent Variables	**Est. Coef.**	**Std. Error**
Intercept	0.227**	0.077
N_{R1}	0.097***	0.017
N_{R1}^2	-0.002**	0.0005
N_{R2}	0.204***	0.056
N_{R2}^2	-0.014*	0.006

***: 0.000 level of signifi. **: 0.001 level of signifi. *: 0.01 level of signifi.

Results of the extensibility test Figure 6.8 shows the converged search process for optimal values. In Case One, the search terminates in the 15^{th} generation as there is no improvement in the best objective value for 15 consecutive generations. Throughout these generations, we identify 18 optimal suction power values, as listed in Table 6.8 corresponding to column $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}, N_{R2}]$, which decrease by 1.248 to 1.336 from the original design and correspond to the best objective value of 11.555. In Case Two, convergence is achieved by the 16^{th} generation. We identify 21 sets of optimal solutions characterized by weight and suction power values $[x_w, x_s]$ given in Table 6.8. These solutions align with the best objective value of 12.525. Comparing these results with the traditional design method and the proposed method before the extension (*i.e.*, not including node role $R2$ into account), the extended method achieves the highest objective values for both cases. For example, in Case Two, highlighted in blue in Table 6.8, the number of times product purchases reaches around 13 when considering both inter-brand and intra-brand competitions. This is approximately three times higher than the traditional design method and five units more than the design that only considers inter-brand competition. This highlights the importance of a comprehensive understanding of the market competition environment for effective product design.



(a) Case One: suction power design case (b) Case Two: suction power and weight design case

Figure 6.8: Iterative search processes using a genetic algorithm to optimize two design cases. The processes terminate after 15 and 16 generations, respectively, with a convergence criterion of no improvement in the best objective value (fitness) for 15 consecutive generations.

Table 6.8: Design Results Comparison: Traditional and Proposed Methods

Traditional Design Method				**Proposed Design Method**					
				$g(y(X)) = [N_{R1}]$			$g(y(X)) = [N_{R1}, N_{R2}]$		
Case One: X = $[x_s]$									
	x_s		u		x_s	u		x_s	u
Original Design	4		2.364[a]	Original Design	4	0.854[a]	Original Design	4	2.146[a]
Optimal Design	5		3.353	Optimal Design	[2.676, 2.680, 2.684, 2.688, 2.692, 2.696, 2.704, 2.708, 2.712, 2.716]	6.493	Optimal Design	[2.664, 2.668, 2.672, 2.676, 2.684, 2.692, 2.696, 2.704, 2.716, 2.720, 2.724, 2.728, 2.732, 2.736, 2.740, 2.744, 2.748, 2.752]	11.555
Case Two: X = $[x_s, x_w]$									
	x_s	x_w	u		$[x_s, x_w]$	u		$[x_s, x_w]$	u
Original Design	4	15.6 LB	2.802	Original Design	[4, 15.6 LB]	0.854[b]	Original Design	[4, 15.6 LB]	2.146[b]
Optimal Design	5	23.781 LB	4.033	Optimal Design	[4.291, 22.317 LB] [4.445, 28.457 LB] [1.985, 20.078 LB]	7.729	**Optimal Design**	**[2.904, 18.723 LB] [3.216, 18.760 LB] [3.252, 18.797 LB] [3.164, 18.723 LB] [3.236, 18.760 LB] [3.280, 18.797 LB] [3.200, 18.723 LB] [3.296, 18.760 LB] [3.316, 18.797 LB] [3.132, 18.760 LB] [3.380, 18.760 LB] [3.348, 18.797 LB] [3.156, 18.760 LB] [3.384, 18.760 LB] [3.408, 18.797 LB] [3.176, 18.760 LB] [2.976, 18.797 LB] [3.164, 19.093 LB] [3.188, 18.760 LB] [3.228, 18.797 LB] [3.372, 19.204 LB]**	**12.525**

[a]: The u value corresponding to the original design is calculated by inputting the original design values of Product 369 into the estimated Negative Binomial Regression model for the traditional design method or into the developed algorithm for the proposed design method.
[b]: Corresponding to a specific $g(y(X))$, case one and case two share the same algorithm, resulting in the same value of u for the original design of both cases.

6.4 Discussion

In this section, we first discuss the generalizability of the proposed network-based design framework, and then discuss the limitations of the current work and suggest future directions for improvement.

Generalizability of the proposed method The generalizability of the proposed method is represented in two aspects: 1) *Generalizability in handling complexity.* The proposed method is flexible and can handle different levels of complexity in optimization design problems. As illustrated in Section 6.3.5, it can optimize a varying number of product attributes. Additionally, Section 6.3.6 demonstrates that the derived local network-based design variable $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X}))$ can be adjusted to include different local network structures. This adaptability allows the model to manage a wide range of complexities, making it applicable to various scenarios. 2) *Generalizability across cases.* Beyond vacuum cleaner product design, the proposed method can be directly applied to other product designs, such as vehicles and cellphones, by incorporating market competition information into the product design process. Additionally, the method can be generalized to guide the design of other networked systems, such as transportation systems and power grids. For example, in a shared mobility system, each docked bike station can be defined as a node in a trip network, with directed links representing trips between stations. Using a network motif mining tool, significant travel patterns can be identified to formulate the derived local network-based design variable for each station. This variable, determined by original station design parameters like dock numbers, can incorporate significant user travel patterns into station capacity design, improving system performance, such as user satisfaction scores.

Limitation The first limitation of the current work is the inadequate predictive accuracy of the ERGM obtained in Step 4. One key reason for this could be attributed to the data insufficiency. As stated in our previous study (Xiao et al., 2023b), the

data for US household vacuum cleaners, including 945 customer responses to 612 unique vacuum cleaner models are quite heterogeneous, *i.e.*, most customers' preferred vacuum cleaners are very different from others. This makes the co-consideration network have insufficient links to train an effective ERGM for prediction. Inspired by the existing study (Ahmed et al., 2022b), where a GNN model was trained using a dataset aggregated from more than 40,000 vehicle survey responses to predict the co-consideration network for the vehicle market system with an $F1\text{-}Score$ of 0.65, we propose two potential solutions. First, we could collect more data. With more customer responses, we believe that the accuracy of the model will be improved. Second, we could use advanced deep learning models to replace ERGM since Step 4 of the proposed methodology only requires a network predictive model, we plan to test more advanced deep learning models such as GNN as the surrogate model that is expected to further improve prediction accuracy.

Another limitation of this study lies in the slow computational efficiency of the Algorithm 1. The computer used in the experiments is equipped with an 11th-gen Intel Core CPU (i9-11900 2.50 GHz, 8 cores, 16 logical processors) and 32GB of RAM. Since ERGM is not compatible with GPU calculation, we employ a parallel computing strategy utilizing 14 logical processors of the CPU. The computational time for each round of Algorithm 1 is approximately 4.3 minutes for only considering the inter-brand competition and 4.8 minutes for considering both inter-brand and intra-brand competition. Consequently, each generation involving 30 populations of the genetic algorithm requires a total of 2.15 and 2.4 hours, respectively. Therefore, solving the proposed optimal design problem and its extended version, which encompasses 15 generations, requires 32.25 hours and 36 hours to complete the calculations. Moreover, the inefficiency of the Algorithm 1 will also hinder the applicability of the proposed method to systems with large network sizes. To address this computational challenge, there are two potential directions to explore. One approach involves extending the current ERGM package (Krivitsky et al., 2023) to make it compatible with GPU computing. Another direction is to utilize the aforementioned GNN model, which is

GPU-compatible, as a replacement for ERGM in calculating the objective value.

6.5 Conclusion

In this study, we introduce a network-based system design framework, consisting of six key steps. The first step involves generating a network representation for the complex systems. In the second step, we perform significant local network mining and articulate the local network-based design goal, which is an essential step in integrating interdependencies between individual entities into the design process. Subsequently, in Step 3, we formulate an optimization problem based on the proposed local network-based design goal. Moving on to the fourth step, we develop a network predictive model as a surrogate to evaluate the system objective to prepare for solving the optimization problem. In Step 5, we integrate the predictive model into optimization algorithms, such as the genetic algorithm, to address the optimization problem outlined in Step 3. Here, the predictive model plays a key role in updating the objective value, while the genetic algorithm is employed to search for the optimal objective value. Finally, the obtained optimal design solution is utilized to recompute the objective value for validation.

To demonstrate the applicability of our approach in real-world scenarios, we present a case study on the US household vacuum cleaner market. The objective is to optimize a specific product model's design attributes to increase its sales. Following the proposed method, we first model the vacuum cleaner market competition as a unidimensional co-consideration network. Next, we employ network motif theory to identify three significant local competition patterns and define the derived local network-based design variable based on the unique node positions in those identified competition motifs. This derived variable is a function of the original product design variables, including suction power, weight, and price (as a constrained variable). With the goal of maximizing the number of times a vacuum cleaner purchases, we formulate a local network-based design objective function by estimating the relation-

ships between purchase times and the derived network-based design variable using negative binomial regression. We then frame an optimization problem based on this objective function and solve it using a typical genetic algorithm procedure. In this process, the ERGM-based predictive model works as a surrogate model to evaluate the system objective whenever a new value of the design attributes (*i.e.*, weight and suction power) is explored.

We demonstrate the efficiency of our proposed design method by comparing it with the traditional design method. The results show that the optimal values of suction power and weight found by the proposed method can significantly enhance the number of times product purchases, achieving about twice the increase compared to the traditional design method. Additionally, we demonstrate the extensibility of the proposed method by modifying the derived design variable to include more competition relations. The highest objective value illustrates the success of this extension and highlights the importance of comprehensively understanding the market competition environment for optimal product design. Finally, the formulation of the micro-level entity design framework, which accounts for meso-level dependencies, aids in addressing ***RQ1***, ***RQ2***, and ***RQ3*** by examining how meso-level subsystems influence the functionality of micro-level individual entities.

Chapter 7: Preliminary Exploration of Socio-Technical Systems Dynamics Based On Meso-Level Significant Temporal Subsystems

7.1 Overview

In this chapter, the objective is to conduct a preliminary exploration of socio-technical system (STS) dynamics, with two sub-objectives. The first sub-objective is to explore potential solutions for dynamic data scarcity in socio-technical systems. However, given the highly imbalanced nature of the STS network data (*i.e.*, sparse network), this study also aims to provide insights into addressing this imbalance challenge by proposing a comprehensive experimental framework based on GNN-based link prediction models. Both SMS and CPMS are employed for validation. The second sub-objective is to develop a meso-level temporal subsystem-based analysis framework for temporal STSs. The development of this framework contributes to answering ***RQ1*** and ***RQ2*** when considering the time dimension. CPMS, specifically the US vehicle market system, is utilized for validation.

In detailed, this chapter is organized as follows:

- Section 7.2 introduces an experimental framework encapsulating the data undersampling method, GNN-based LP models, model post-processing methods, and result evaluation strategies.
- Section 7.3 introduces the proposed meso-level temporal subsystem-based analysis framework of temporal STSs, including network modeling of dynamic STSs and temporal network motif mining and interpretation.
- Section 7.4 concludes this chapter with closing thoughts.

7.2 Graph Neural Network-Based Link Prediction (LP) for Highly Imbalanced Network Data

7.2.1 Experiment Framework

Figure 7.1 provides an overview of the experimental framework, encapsulating the data undersampling method, GNN-based LP models, model post-processing methods, and result evaluation strategies. The following sub-sections delve into each stage, unraveling the details from data sampling to result evaluation.

7.2.1.1 Data Undersampling Method

A significant challenge for LP lies in the presence of too few positive links, leading to a severely imbalanced binary classification problem. One prevalent method to tackle this issue involves undersampling the majority class to achieve balance in training data (Brownlee, 2020). Illustrated in Figure 7.2, the initial step of the undersampling process for LP involves obtaining the link set E, encompassing all possible links given the node set V within a training network G. For instance, if set V comprises $N_V = 10$ nodes, the total number of possible undirected links in V is $N_V(N_V - 1)/2 = 45$ (for directed links, the total number is $N_V(N_V - 1)$). Subsequently, all links are labeled according to their existence in the network, G. For example, in the instance of network G depicted in Figure 7.2, the link between node 1 and node 2 is observed, thus labeled "1", while the link between node 1 and node 4 does not exist and hence is labeled "0."

After labeling, E is divided into two subsets: E_p, which preserves all positive

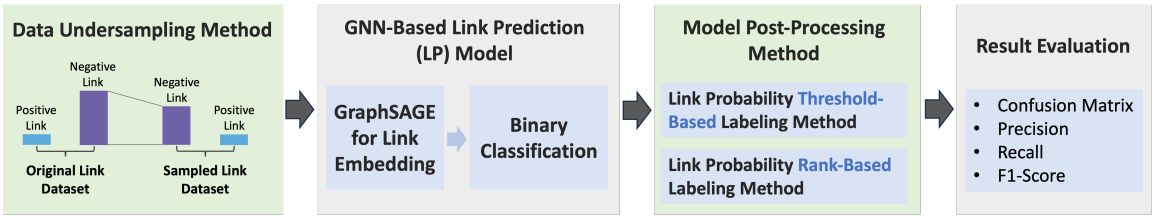


Figure 7.1: Experiment framework.

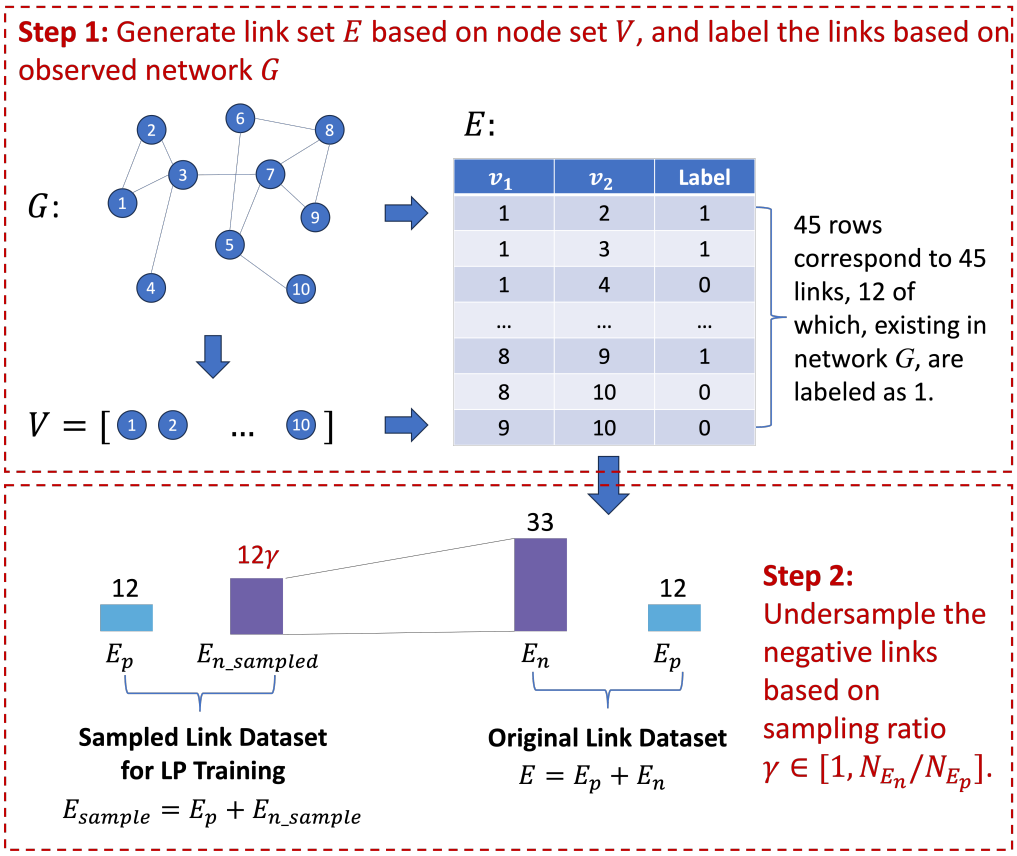


Figure 7.2: Undersampling process.

links, and E_n, which stores all negative links. Finally, the negative links are randomly sampled with respect to the given ratio $\gamma \in [1, N_{E_n}/N_{E_p}]$, where $\gamma = 1$ indicates that the sampled link set includes an equal number of positive and negative links. Conversely, $\gamma = N_{E_n}/N_{E_p}$ signifies no sampling process is carried out and all negative links are selected for training. Therefore, the γ value indicates the degree of imbalance within the training dataset. In this work, we are especially interested in how different data undersampling ratios (*i.e.*, the γ values) influence prediction results with a given GNN-based LP model.

7.2.1.2 GNN-Based Link Prediction (LP) Model

In this study, we adhere to our previously proposed GNN-based LP model, as detailed in Chapter 4. Illustrated in Figure 4.3, the model architecture comprises two principal components: the first involves utilizing the GraphSAGE algorithm (Hamilton et al., 2017) for link embedding, while the second relates to binary classification for link prediction. In the first component, we employ the GraphSAGE algorithm (Hamilton et al., 2017) to generate a new vector representation of size M for each node, referred to as *node embedding*. This representation is derived by aggregating each node's individual features alongside its two-hop network neighborhood information. It is important to note that the model shown in Figure 4.3 was created for directed networks. For undirected networks, the distinction between in- and out-neighbors is not applicable, and the network neighbors are treated uniformly. Additionally, training GraphSAGE necessitates providing network information for each node to aggregate neighborhood data, which can be approximated using the K-nearest neighbor method (Ahmed et al., 2022b) or a regular artificial neural network (Xiao et al., 2023a).

Once individual node embeddings are obtained, they are concatenated to form a *link embedding*. In contrast to the directed network's link embedding shown in Figure 4.3, the undirected version is insensitive to the order of the start- and end-nodes. Moving on to the second component, the resultant link embedding of size $2M$ is fed into a fully connected neural network with one hidden layer. Both the input and hidden layers share the same size as the link embedding, and the output layer comprises a single neuron that utilizes the Sigmoid activation function. This output layer produces a probability of the input link's existence. The entire training process follows an end-to-end supervised learning approach, with the aim of minimizing binary cross-entropy loss using stochastic gradient descent (SGD) (Nielsen, 2015).

7.2.1.3 Model Post-Processing Methods

In this study, we introduce two model post-processing methods that convert the probability of link existence into binary labels. The first method is the threshold-based labeling method, and the second is the rank-based labeling method.

Link probability threshold-based labeling As depicted in Figure 7.3, the probability threshold-based labeling method begins by establishing an optimal threshold, denoted as $P_{\text{threshold}}$. Then, the predicted probabilities undergo a transformation into labels using the rule: links with probabilities higher than $P_{\text{threshold}}$ are labeled "1" while those below the threshold are labeled "0."

Once all labels are created, a confusion matrix (Nielsen, 2015) of this binary classification can be obtained from which we are able to compute the main accuracy metrics, such as true positive rate (recall) and precision. In this study, our focus is primarily on the model's proficiency in predicting minority (positive) links. Consequently, we adopt the optimal point on the precision-recall (PR) curve, specifically the point with the highest F1-Score. The PR curve represents the plot of precision versus recall at various thresholds of link probability. The F1-Score, expressed by Equation (7.1), is the harmonic mean of precision and recall, providing a balanced evaluation metric (Xiao et al., 2023a). The probability threshold-based method emphasizes the optimal trade-off between precision and recall by setting a specific threshold value.

$$F1\text{-}Score = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{7.1}$$

Link probability rank-based labeling In the rank-based labeling method, we commence by ranking the predicted probabilities from high to low, as illustrated in Figure 7.3. Here, we introduce the hit ratio at the top-K ranked links ($HR@K$), a metric commonly employed in recommendation systems to calculate the recall rate (Wu et al., 2022). The formula for $HR@K$ is given by:

$$HR@K = \frac{N_p^K}{N_{E_p}}, \tag{7.2}$$

where N_p^K represents the number of true positive links included in the top-K ranked links, and N_{E_p} is the total number of true positive links in the test dataset. For this study, since we aim to keep the predicted network with the same density (N_{E_p}/N_E) as the ground-truth network, we set $K = N_{E_p}$[1]. To illustrate, in Figure 7.3, we set $K = 4$, which means the observation of 4 true positive links in E_{test}. The $HR@4$ in the lower-right plot of Figure 7.3 is 75%, indicating that the links with the top-4 probabilities can correctly predict 75% of true positive links. In simpler terms, if we label these top-K links as "1" and the rest as "0," we can achieve a recall of 75%. In contrast to the probability threshold-based method, which seeks a balance between precision and recall, this proposed rank-based method places greater emphasis on analogizing the size of the observed network.

7.2.1.4 Result Evaluation

In this study, we employ widely recognized metrics for binary classification evaluation. We initiate the evaluation process by comparing the predicted labels with the true labels to obtain the confusion matrix. Subsequently, we calculate key metrics, including the true negative rate (TNR), false positive rate (FPR), true positive rate (TPR), false negative rate (FNR), and precision (which is equivalent to $TP/(TP + FP)$). The F1-Score is then computed using Equation (7.1), with values ranging from 0 to 1. A higher F1-Score indicates a superior performance of an LP model (Brownlee, 2020).

[1]It is worth noting that the rank-based labeling method is a general method. While density serves as an illustrative example, other metrics of interest, such as hit rate (recall), can also be seamlessly used to determine the top-K.

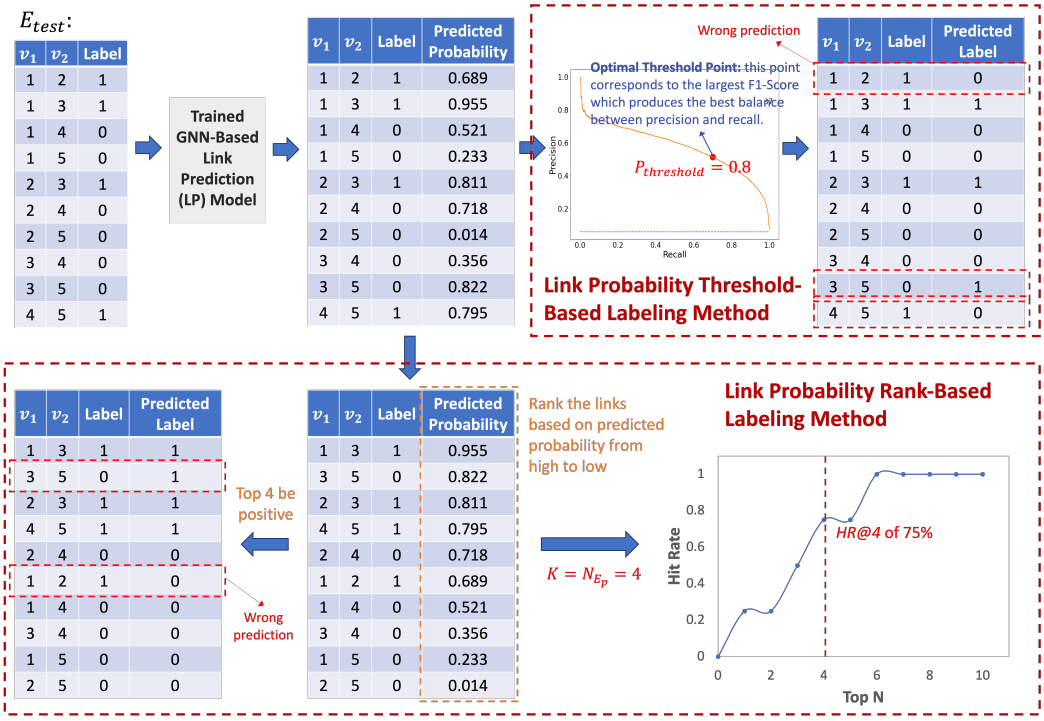


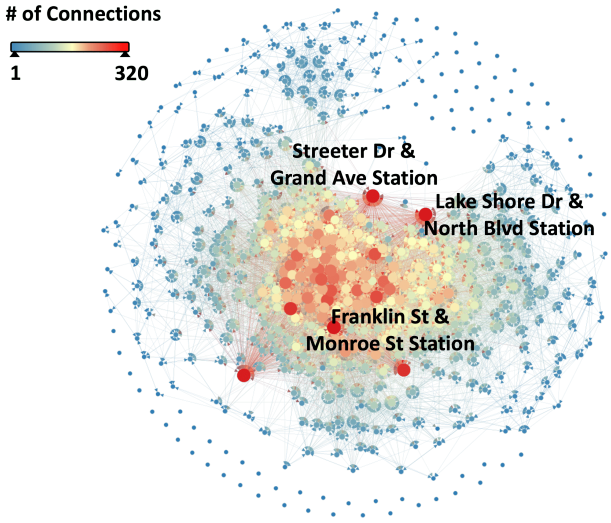
Figure 7.3: Illustration of model post-processing methods.

7.2.2 Data Source and Experiment Preparation

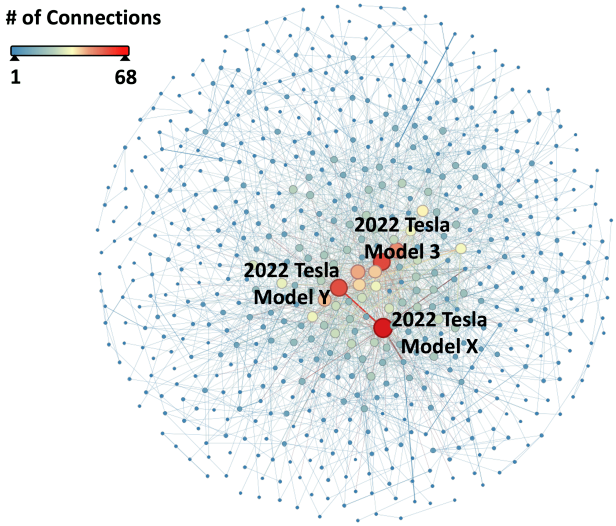
In this section, we applied two datasets to conduct the experiment. The first dataset pertains to the shared mobility system, and the second dataset relates to the vehicle market system. We outline the detailed experiment settings corresponding to these two test cases below.

7.2.2.1 Data Source

Divvy Bike shared mobility system data For the shared mobility system data, we continue to use the binary directed trip network data introduced in Section‘4.3.1, where the network structure is illustrated in Figure 7.4 (a). This network comprises 535 station nodes, each characterized by ten features, including two geographic coordinate features, one capacity feature (number of docks), and seven POI features. Out of the 285,690 potential directed links, 7.4% (21,221 positive links) are observed.



(a) 2016 Binary Directed Trip Network
(# of Nodes: 535, # of Links: 21221)



(b) 2023 Binary Undirected Co-consideration Network
(# of Nodes: 624, # of Links: 2151)

Figure 7.4: Visualisations of the training networks.

US vehicle market survey data This dataset originates from our recent survey study on US new car buyers, which comprises two parts. The first part encompasses car attribute data, selecting 624 distinct car models with a total of 22 features (*e.g.*, brand, fuel economy, base curb weight, etc.) from the dataset collected in Section 5.3.1. The second part involves data from responses from new car buyers, where participants provided information on up to six cars they considered, including model year and name.

In total, we collected consideration sets from 2,283 respondents. We constructed a binary undirected vehicle co-consideration network following the definition of the co-consideration network (Xiao and Cui, 2023). So, each node represents a unique vehicle model, and a link is established between two vehicles if they are co-considered by at least one buyer. The visualization of the resulting co-consideration network is presented in Figure 7.4 (b), featuring 624 nodes and 2,151 links, constituting 1.1% of all 194,376 possible undirected links.

7.2.2.2 Design of Experiment

The detailed experimental scheme is presented in Table 7.1, encompassing a total of 20 tests. To enhance consistency and minimize variability, we employ the cross-fold validation approach (Bengio and Grandvalet, 2003) for each test. Taking the shared mobility system as an illustration, we divide the 535 nodes into five folds, each comprising 107 nodes. Subsequently, we iteratively designate one of these folds for validation and employ the remaining four folds for training. During this process, for both training and validation link sets, only the links that are observed in the original network in Figure 7.4 (a) are labeled as "1" and the rest as "0." This procedure yields five original link sets (before applying the undersampling strategies) for training and five validation link sets. It is important to note that those original link sets for training are rebalanced according to a specified undersampling ratio before being fed into the model for each test. Table 7.3 and Table 7.4 provide detailed statistics for

each original set for training and validation link set for both cases.

Table 7.1: Experiment Scheme.

Shared Mobility System

Undersampling Ratio γ	**Post-processing Method**
1, 3, 5, 8[1], No sampling[2]	Link probability *threshold-based* labeling method[3]
1, 3, 5, 8, No sampling	Link probability *rank-based* labeling method ($K = 7.4\% * N_E^{val}$)[4]

Vehicle Market System

Undersampling Ratio γ	**Post-processing Method**
1, 3, 5, 50[1], No sampling	Link probability *threshold-based* labeling method
1, 3, 5, 50, No sampling	Link probability *rank-based* labeling method ($K = 1.1\% * N_E^{val}$)

[1]: These two values are chosen in each test case because they are the nearest integer to the median of the range $[1, N_{E_n}/N_{E_p}]$.
[2]: No sampling implies that we include all positive and negative links from the observed network (ground truth) as the training dataset.
[3]: The optimal probability threshold for each test is determined in the post PR curve analysis, as detailed in the results section.
[4]: N_E^{val} represents the total number of all potential links in the validation set.

Next, as outlined in Section 7.2.1.2, the training of the GraphSAGE LP model necessitates a reference network for the aggregation of network neighborhood information. In this study, to mitigate the potential consequences of inaccurate embedding of neighborhood information during experiments, we opt to directly input the original network, as depicted in Figure 7.4, into the LP model for both systems. Finally, we summarize the model settings and hyperparameter values in Table 7.2.

7.2.3 Experiment Results

7.2.3.1 Confusion matrix

In this section, we examine the results of the confusion matrices of all tests conducted on both the shared mobility system and the vehicle market system, which are summarized in Figure 7.5. As mentioned above, a five-fold cross-validation approach is

Table 7.2: Experiment parameter settings.

System	Setting Items	Value
Shared Mobility System[1]	Neighborhood search depth	2
	# of Sampled in- and out- neighbors in two hops	10
	Node embedding size	30
	Input and hidden layer size for GraphSAGE	60
	Minibatch size	192
	Learning rate	4e-4
	Dropout	0
	Epoch	200
Vehicle Market System[2]	Neighborhood search depth	2
	# of Sampled neighbors in two hops	5
	Node embedding size	10
	Input and hidden layer size for GraphSAGE	10
	Minibatch size	32
	Learning rate	5e-4
	Dropout	0.3
	Epoch	500

[1]: We adopted these settings from our prior work (Xiao et al., 2022a), with the only alteration being the reduction of epochs from 500 to 200 to improve computational efficiency. As the primary objective of this paper is not to identify the optimal model performance, this epoch setting proves adequate for achieving convergence because a typical decrease is not observed beyond 200 epochs of training.

[2]: These settings are determined by trial and error. We guarantee the model's convergence. Similarly, we prioritize computational efficiency, since our focus does not extend to achieving the best model.

employed for each test, and thus the figures illustrate the mean and standard deviation of the results over five rounds. Additionally, for the threshold-based labeling method, the probability thresholds vary across different tests due to the optimization process involved, as explained in Section 7.2.1.3. The purpose aim is to compare the best performance of each test at the optimal threshold point with the largest F1-Score. The results reveal several insights in the following.

1) Regardless of a system exhibiting moderate or extreme data imbalance, the threshold-based labeling method consistently achieves higher recall, indicating better identification of true positive links compared to the rank-based method.

Table 7.3: Cross-fold training and validation data statistics for shared mobility system.

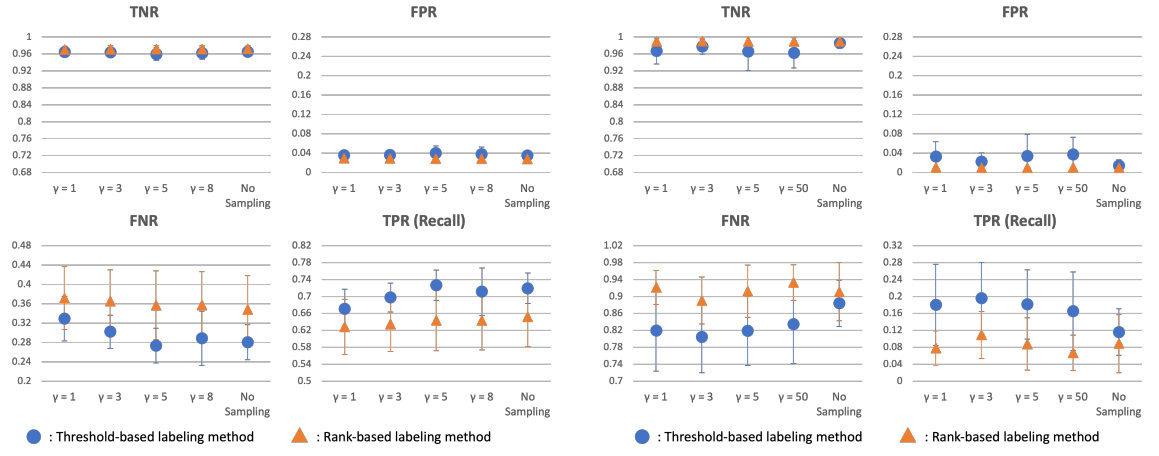
Original Link Set for Training	**# of Nodes**	**# of Positive Links (N_{E_p})**	**# of Negative Links (N_{E_n})**
1	428	12,647	170,109
2	428	13,345	169,411
3	428	14,528	168,228
4	428	14,642	168,114
5	428	12,761	169,995
Validation Set	**# of Nodes**	**# of Positive Links (N_{E_p})**	**# of Negative Links (N_{E_n})**
1	107	1,077	10,265
2	107	887	10,455
3	107	663	10,679
4	107	598	10,744
5	107	1,035	10,307

Table 7.4: Cross-fold training and validation data statistics for vehicle market system.

Original Link Set for Training	**# of Nodes**	**# of Positive Links (N_{E_p})**	**# of Negative Links (N_{E_n})**
1	500	1,253	123,497
2	500	1,482	123,268
3	500	1,369	123,381
4	500	1,411	123,339
5	496	1,379	121,381
Validation Set	**# of Nodes**	**# of Positive Links (N_{E_p})**	**# of Negative Links (N_{E_n})**
1	124	117	7,509
2	124	74	7,552
3	124	91	7,535
4	124	76	7,550
5	128	83	8,045

However, this improvement comes at the cost of predicting more false positives, as evidenced by a higher FPR and a lower TNR.

2) In the case of the shared mobility system with a moderate imbalance issue, a t-test is conducted to assess the observed increasing trend of recall in both



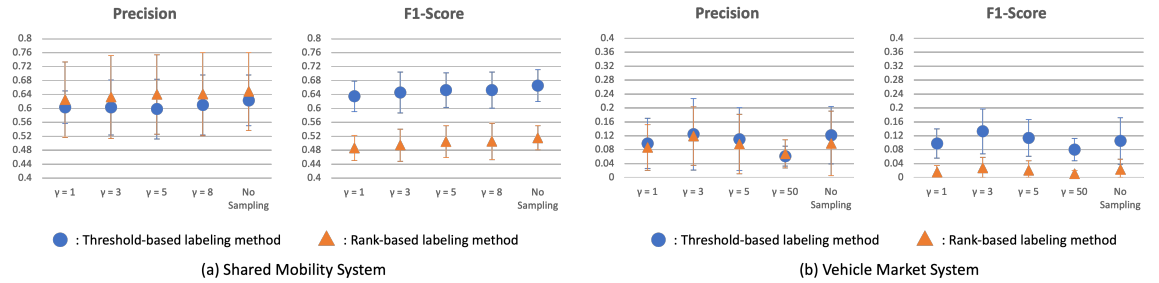
Optimal probability threshold for threshold-based labeling method				
$\gamma = 1$	$\gamma = 3$	$\gamma = 5$	$\gamma = 8$	No Sampling
0.846	0.701	0.514	0.429	0.345

(a) Shared Mobility System

Optimal probability threshold for threshold-based labeling method				
$\gamma = 1$	$\gamma = 3$	$\gamma = 5$	$\gamma = 50$	No Sampling
0.768	0.617	0.447	0.046	0.046

(b) Vehicle Market System

Figure 7.5: Confusion matrix results statistics (mean ± standard deviation) for all tests conducted on both systems.



(a) Shared Mobility System

(b) Vehicle Market System

Figure 7.6: Precision and F1-Score statistics (mean ± standard deviation) of all tests for both systems.

labeling methods. Our null hypothesis posits that there is no increase in recall. The resulting p-values for the threshold-based and rank-based methods are far less than 0.001. Hence, the hypothesis is rejected, Consequently, including more negative links (that is, as the γ value increases) is beneficial for increasing recall in both labeling methods, and these increases are statistically significant. Conversely, in a system with a more serious imbalance issue, the inclusion of more negative links may lead to a decrease in recall, especially for the threshold-based labeling method.

3) The exacerbated imbalance issue contributes to more dispersed performance, as reflected in the larger standard deviation observed in the plots for the vehicle market system. One potential reason could be attributed to the scarcity of positive samples relative to the overwhelming number of negative samples, resulting in greater variability in the model's predictions.

7.2.3.2 Precision and F1-Score

Furthermore, upon examining the Precision and F1-Score results, we provide additional evidence that, in systems with moderate imbalance issues, increasing the undersampling ratio enhances the overall model performance, leading to a higher F1-Score. A t-test is conducted, and the resulting p-value is less than 0.001, indicating the statistical significance of this increase. This positive effect is observed for both labeling methods. The potential explanation lies in the increased information provided to the LP model when augmenting the number of negative links. In systems with more extreme imbalance data, one interesting observation is that an optimal undersampling ratio ($\gamma = 3$) appears to exist, particularly enhancing the predictive performance of the threshold-based method, resulting in the highest F1-Score. However, when no sampling method is applied, the vehicle market system exhibits the highest TNR and precision, along with the lowest recall. This low recall could be attributed to the escalating dominance of negative samples and the scarcity of positive samples, causing the LP model to exhibit bias toward predicting more samples as negative. Therefore, when dealing with large and sparse networks, implementing the proposed undersampling ratio strategy is recommended to determine the optimal γ, achieving the best trade-off between computational efficiency and model performance.

Lastly, the proposed rank-based labeling method exhibits inferior performance compared to the threshold-based method for both systems. This discrepancy is attributed to the distinct objectives of the two methods. The threshold-based method aims to achieve the best F1-Score, while the rank-based method strives to analo-

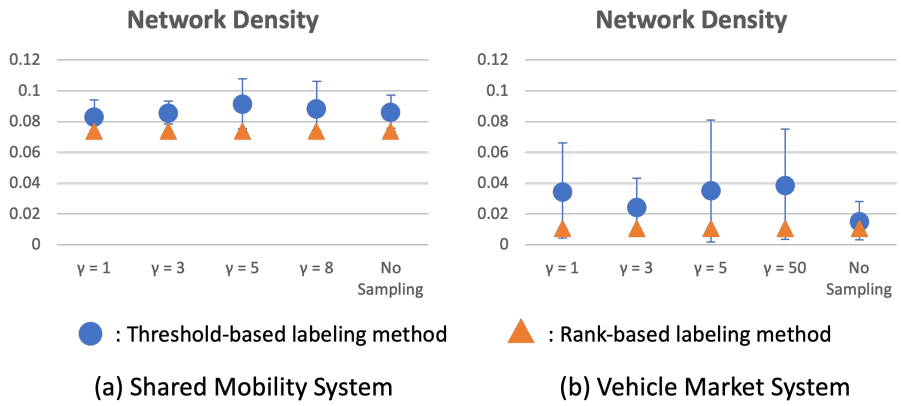


Figure 7.7: Predicted network density statistics (mean ± standard deviation) of all tests for both systems. The density here represents the ratio between the number of predicted positive links and the total number of possible links.

gize the network size of the system. Consequently, the rank-based method tends to be conservative in predicting positive links, leading to higher precisions than the threshold-based method in the shared mobility system. This is visually represented in Figure 7.7, where it is apparent that the threshold-based method predicts more positive links for both systems, resulting in a higher density. Particularly for the vehicle market system, the mean network density for the threshold-based method is on average three times higher than that of the rank-based method.

In summary, for a system characterized by moderately imbalanced network data, increasing the undersampling ratio proves effective in enhancing predictive performance, resulting in a larger F1-Score. Conversely, this strategy is not applicable to systems with extremely imbalanced network data. Moreover, in a broader context, the threshold-based labeling method consistently outperforms the proposed rank-based method. The rank-based approach aims to predict a network of comparable size to the original observed network but sacrifices a portion of prediction accuracy in the process.

7.3 Meso-Level Temporal Subsystem-Based STS Dynamic Analysis

7.3.1 Meso-Level Temporal Subsystem-Based Analysis Framework of Temporal STSs

The proposed meso-level temporal subsystem-based analysis framework of temporal STSs includes two major steps where the first step is network modeling of dynamic STSs and the second step is temporal network motif mining and interpretation. Each step is introduced in detail below.

7.3.1.1 Dynamic Network Modeling

As illustrated in Figure 7.8, the process of dynamic network modeling initiates with the generation of two-year co-consideration networks, following the same definition as outlined in Chapter 6. Subsequently, these two networks are synthesized into one, retaining only the nodes present in both years and the links associated with these nodes that either manifest in Year 1 or Year 2. Ultimately, the links within the synthesized dynamic network are color-coded based on their existence status. As depicted in the right-most block of Figure 7.8, links exclusively present in Year 1 are represented in orange dotted lines, those appearing in both Year 1 and Year 2 are denoted in blue solid lines, and links newly emerging in Year 2 are highlighted in green dashed lines. Consequently, the colored network allows for the observation of three distinct types of links. This simplified discrete representation of temporal information offers several benefits. Firstly, it facilitates preliminary descriptive exploration of the dynamic evolution of socio-technical systems (STSs) (Skarding et al., 2021). Additionally, this representation streamlines the temporal subsystem mining process by seamlessly integrating with existing static motif mining tools.

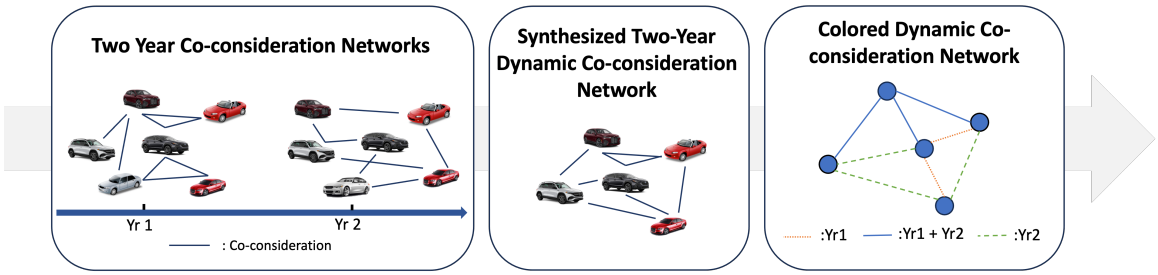


Figure 7.8: Dynamic network modeling.

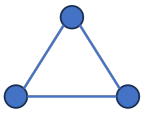
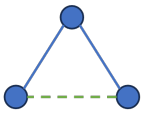
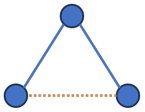
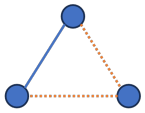
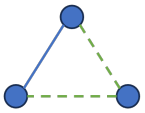
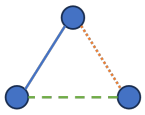
7.3.1.2 Temporal Network Motif Mining and Interpretation

Upon obtaining the colored dynamic co-consideration network, it is input into the motif mining tool (*e.g.*, FANMOD (Rasche and Wernicke, 2006)) to identify statistically significant temporal competition motif patterns. Within the defined dynamic co-consideration network context, there are ten temporal motifs with closed forms and unique structures. This study focuses on patterns that include at least one link persisting in both Year 1 and Year 2 (blue solid link). These patterns are detailed in Table 7.5, arranged in descending order based on the number of blue solid links. Furthermore, each pattern is empirically interpreted within the context of co-consideration networks. This analysis offers valuable insights into the dynamic competitive landscape among products, enriching our comprehension of consumer preferences and market dynamics. For instance, companies can identify long-term competitors by examining the TCM-1 patterns involving their products and assess potential declines in competitiveness by monitoring their involvement in disadvantageous positions in TCM-4.

7.3.2 Case Study: US Vehicle Market System

In this section, we use the US vehicle market system as a case study to illustrate the implementation of the proposed framework in Section 7.3.

Table 7.5: Empirical interpretation of the interested temporal competition motifs.

ID[1]	Structure	Adjacency Matrix[2]	Interpretation
TCM-1		0 2 2 2 0 2 2 2 0	Three products maintain consistent co-consideration relations across both years, indicating stable or enduring competitions over time.
TCM-2		0 2 2 2 0 3 2 3 0	The shared presence of a common competitor between two products, along with sustained competition over consecutive years, serves as the driving force behind they competing with each other.
TCM-3		0 2 2 2 0 1 2 1 0	Two products maintained continuous joint consideration over both years, while the third product only shared association with one of them in the initial year, indicating shifts in consumer preferences or market dynamics over time.
TCM-4		0 2 1 2 0 1 1 1 0	One product is losing market competitiveness in local competition due to its inability to sustain ongoing competitive relations.
TCM-5		0 2 3 2 0 3 3 3 0	A new competitor is emerging to reshape the local competitive relations.
TCM-6		0 2 1 2 0 3 1 3 0	This pattern suggests dynamic shifts in competition over time among the products, with one consistently competing with another, while the third alternates between competitive engagement with the others.

[1]: TCM is Temporal Competition Motif in short.
[2]: In the adjacency matrix, 1 represents the link emerges in Year 1; 2 represents the link emerges in Year 1 and Year 2; 3 represents the link emerges in Year 2.

7.3.2.1 Data Source

The US vehicle market survey data discussed in Section 7.2.2.1 comprises single-year data, limiting its capacity for dynamic analysis. To overcome this limitation, we utilize multiple-year survey data from Ford Motor Company spanning 2017 to 2022. Figure 7.9 presents statistics on respondent numbers and unique car model purchases each year. For instance, in 2017, there were 161,580 respondents purchasing 430 unique car models. However, a notable drawback of this dataset is its lack of customer

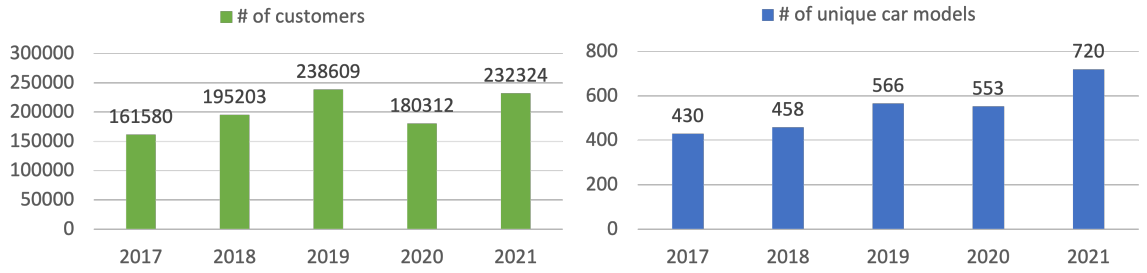


Figure 7.9: The statistics of the multiple-year US new car buyer survey data.

consideration sets, rendering the generation of co-consideration networks impossible. To mitigate this challenge, we employ the LP model trained in Section 7.2 to predict co-consideration relationships among car models annually. Further details on the prediction process are presented in the subsequent section.

7.3.2.2 Dynamic Co-Consideration Network Modeling

Predicting co-consideration networks across multiple years To strike a balance between network density, aiming for comparability with the reference (Ahmed et al., 2022a), and achieving a satisfactory true positive rate, we selected an LP model for prediction. This model utilizes a rank-based labeling method without sampling, targeting a true positive rate of 70%. However, the resulting F1-Score is only 0.062. This low score can be attributed to the persistent challenge of highly imbalanced data, as discussed in Section 7.2. The imbalance issue often leads the LP model to misclassify true negative links as positive, thereby lowering precision. Resolving this imbalance remains a significant challenge requiring further investigation. Nevertheless, it is crucial to recognize that the primary objective of this study is to validate the proposed dynamic analyzing framework. Thus, despite the challenges, the focus remains on demonstrating the efficacy of our approach in the context of predicting co-consideration networks.

The prediction results are illustrated in Figure 7.10. Nodes depicted with larger sizes and darker colors indicate higher popularity in the market, as they are co-considered with a greater number of unique models. Additionally, the most popular

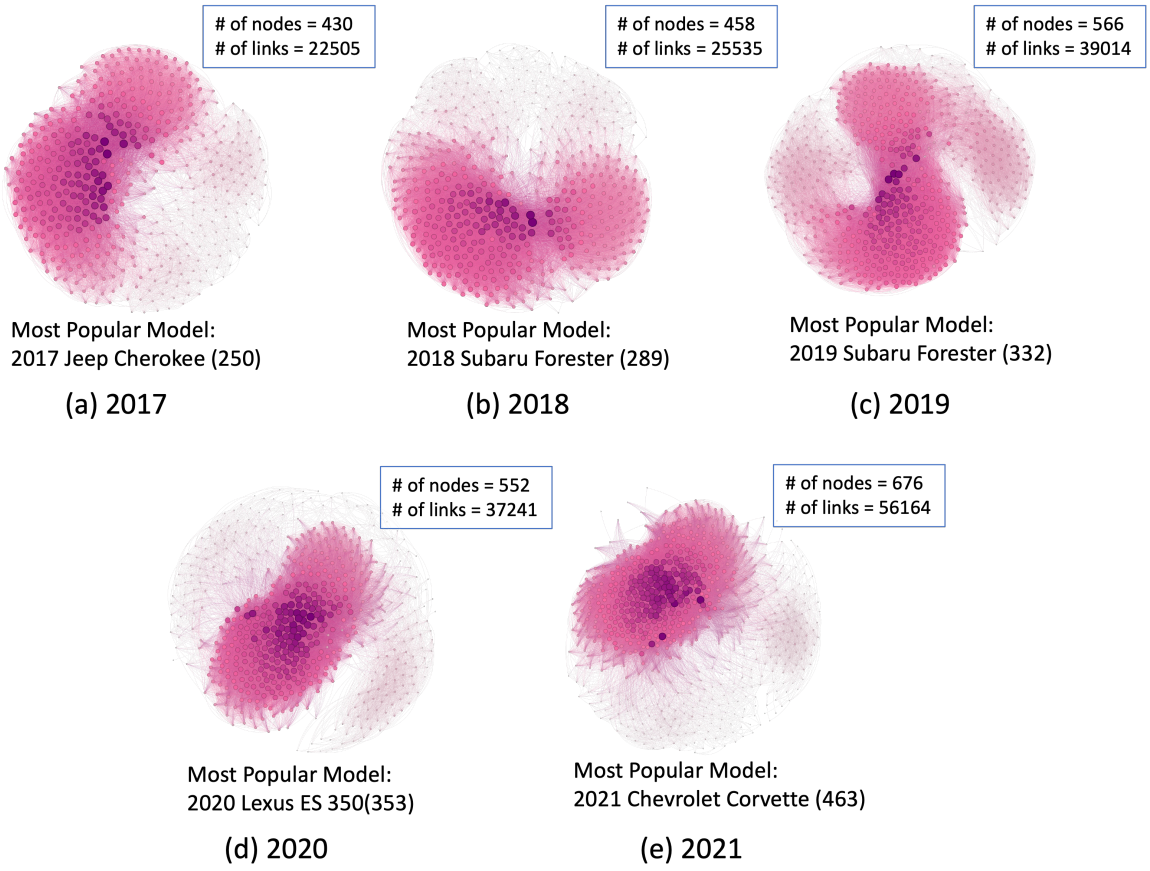


Figure 7.10: Predicted co-consideration networks across 2017 to 2021.

model for each year, along with the number of its connections, is listed beneath each network. For instance, in 2017, Jeep Cherokee was predicted as the most popular car, connecting with 250 other unique car models. As previously mentioned, we regulate network density by adjusting the LP model post-processing method to achieve a target $HR@K = 70\%$. Consequently, these five networks exhibit similar network densities, approaching 0.24.

Dynamic co-consideration networks After obtaining the co-consideration network for each individual year, we synthesize pairs of consecutive networks and apply the link coloring method described in Section 7.3.1.1. Consequently, four temporal networks are generated: 2017 to 2018, 2018 to 2019, 2019 to 2020, and 2020 to 2021.

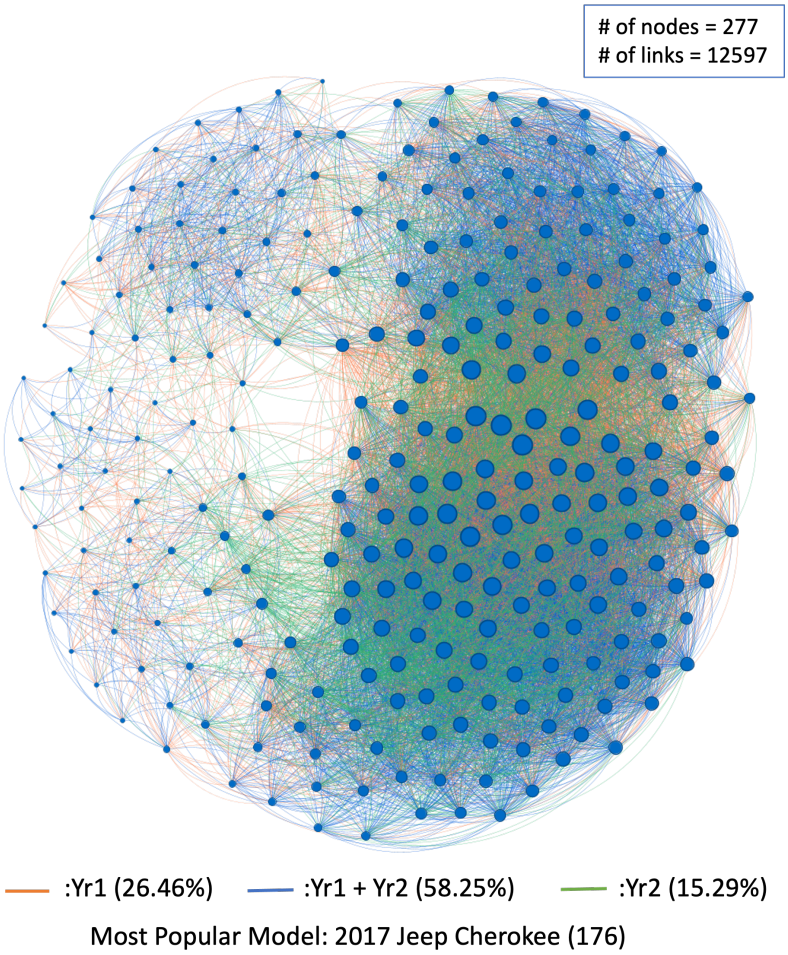


Figure 7.11: An example of the dynamic co-consideration networks (2017 to 2018).

An illustration of the temporal networks is provided in Figure 7.11. In this example, we observe that 277 unique car models maintained their presence in the market from 2017 to 2018, representing approximately 45% of the total 611 unique models across both years. Among these models, 12,597 co-consideration relationships were formed, with 58.25% persisting in both years, 26.46% disappearing in the second year, and 15.29% emerging in the second year. Notably, the 2017 Jeep Cherokee emerges as the most popular model in this dynamic network, co-considered with 176 other models.

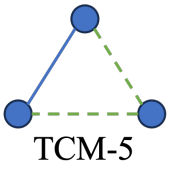
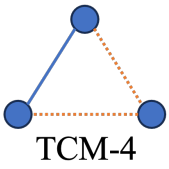
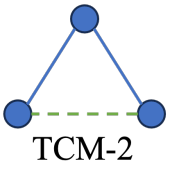
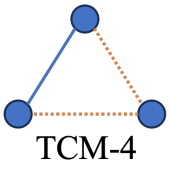
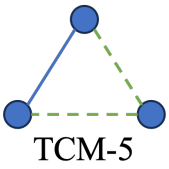
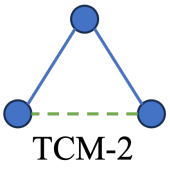
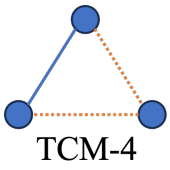
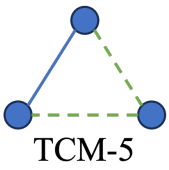
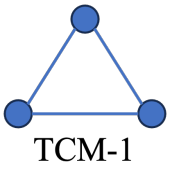
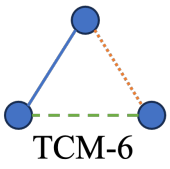
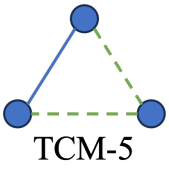
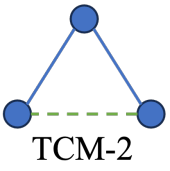
7.3.2.3 Temporal Competition Motif Mining

For the temporal competition motif mining, we employ FANMOD. Table 7.6 presents the statistics of the top three significant TCMs in the aforementioned four dynamic co-consideration networks. Analyzing the table empirically, we observe a trend: from top to bottom, the most significant TCMs transition from embracing new competitors to revealing the loss of competitiveness among existing products. Eventually, we observe a heightened dynamism in local competition between 2020 and 2021. Moreover, the consistent emergence of TCM-5 in the top two significant positions suggests that embracing new products to reshape local competitive relations is a recurring phenomenon across these five years.

7.3.3 Discussion

In this section, we introduce a meso-level temporal subsystem-based analysis framework for temporal STSs, contributing empirical interpretations for six temporal competition motifs. We employ the US vehicle market system as a case study to demonstrate the implementation of both the framework and the interpretations. However, two limitations necessitate further investigation. Firstly, the predictive model exhibits low accuracy, indicated by the low F1-Score. This poor performance is primarily attributed to the highly imbalanced network data of the market, and we have dedicated efforts to exploring possible solutions in the previous section. Nevertheless, our objective in this section is to provide guidance on conducting dynamic analysis based on temporal network motifs. Even though we employ a suboptimal model for predicting multiple years' networks, the insights generated are sufficiently generalizable for future use. Secondly, our analysis in this section concludes with an empirical examination of the temporal competition motifs. Further exploration is needed to quantitatively understand these patterns and integrate meso-level temporal competition information into the product design process. This direction will also guide our future research endeavors.

Table 7.6: The statistics of top-3 significant TCMs in each dynamic co-consideration networks.

Year	Top-3 Significant Temporal Competition Motifs[1]		
2017 to 2018	 TCM-5	 TCM-4	 TCM-2
2018 to 2019	 TCM-4	 TCM-5	 TCM-2
2019 to 2020	 TCM-4	 TCM-5	 TCM-1
2020 to 2021	 TCM-6	 TCM-5	 TCM-2

[1]: The significance of TCM, based on Z-score, decreases from left to right.

7.4 Conclusion

In this chapter, we have made significant contributions to the preliminary exploration of STS dynamics based on meso-level significant temporal subsystems, addressing two fundamental challenges associated with this topic. The first challenge pertains to dynamic data scarcity in STS. Temporal data collection, particularly for complex systems like market systems, is time-consuming and costly. An accurate predictive model becomes imperative to forecast STS evolution when data availability is limited. However, the highly imbalanced nature of STS network data, characterized by sparse

networks, presents a formidable challenge in developing high-performance LP models. To tackle this issue, we introduced a comprehensive experimental framework aimed at investigating the influence of data undersampling with varying sampling ratios and model postprocessing methods on prediction outcomes using a GNN-based LP model. Furthermore, we utilized SMS and CPMS, two highly imbalanced real-world STS network datasets, to underscore the impact of imbalance degree on prediction performance. This framework offers valuable insights into optimizing the performance of GNN-based link prediction models through data sampling and postprocessing methods, without altering the model architecture. It contributes to a broader understanding of effective strategies for managing highly imbalanced datasets in real-world STSs.

The second challenge concerns the lack of understanding of temporal subsystem characteristics and their effects on system evolution. To address this gap, we proposed a meso-level temporal subsystem-based analysis framework, encompassing dynamic network modeling, significant temporal subsystem mining, and empirical interpretation of identified patterns. We demonstrated the efficiency of this framework using the US vehicle market system as a case study. The insights gained into the dynamic characteristics of temporal competition motifs are invaluable for companies seeking to comprehend their products' competitiveness in dynamic market environments. Moreover, this initial endeavor lays the groundwork for subsequent advanced analyses, such as quantifying a product's sustained competitiveness. Ultimately, the proposal of this framework contributes to addressing ***RQ1*** and ***RQ2*** of this dissertation from a temporal perspective.

Chapter 8: Conclusion and Future Works

This chapter provides a summary of the completed research and contributions of this dissertation, identifying the research challenges inherent in the study of network-based STS engineering and design. Finally, these challenges prompt considerations for future research opportunities.

8.1 Conclusions and Contributions

This dissertation aims at finding an efficient method to fundamentally understand the complex interactions between social and technical aspects, and thereby better engineering and designing STSs. Inspired by existing studies that have proven subsystems have significant impacts on system performance across broader domains, the **central hypothesis** of this dissertation is that the meso-level statistically significant connections of individual entities, embedding collective behaviors of entities, are crucial function units of STSs, influencing both macro-level performance and micro-level interactions, and thus deserve scientific investigation in STS engineering and design. To gain a deep understanding of what meaningful subsystem information at the meso-level is and how it can be extracted and used to guide the design of an STS system to achieve the desired system performance, the research objective of this dissertation is to develop a novel meso-level network-based framework for STS engineering and design. This dissertation is driven by answering three *research questions*:

- ***RQ1:*** *How can significant meso-level system structures be identified?*
- ***RQ2:*** *What are the influences of the significant meso-level subsystems on the system performance at the macro level and the interaction mechanism at the micro level?*

- ***RQ3:*** *How can meso-level structural information be used to design an STS to achieve desired macro-level performance and micro-level functionality?*

This dissertation has answered these three research questions from three aspects:

- The first aspect is understanding what influences the meso-level subsystems on the macro-level system performance and how to consider this influence during STS optimal design to achieve desired macro-level performance. The SMS, a representation of networked large service systems, is taken as the case study for investigation. Accordingly, two major research tasks are accomplished. The first one is optimizing STS capacity planning based on network motifs to make the system robust enough against seasonal demand fluctuations. The second task is building a high-performance GNN-based link prediction (LP) model and implementing it to support the validation of STS design decisions.

- The second aspect is learning what influences the meso-level subsystems on the micro-level individual entities' functionalities and how to consider this influence during an entity's optimal design to achieve desired micro-level functionality. The CPMS, specifically the US household vacuum cleaner market system, is taken as the case study for investigation. Accordingly, two research tasks are accomplished. The first one is associated with employing information retrieval and survey design to address the STS data scarcity problem. The second one is applying the identified subsystems to quantify the micro-level individual entity performance and using the generated quantification to guide the optimal design of the individual entity.

- The third aspect is understanding the impacts of meso-level subsystems on the macro-level system performance from the time dimension, *i.e.*, both the macro-level system and meso-level subsystems are equipped with time information.

The CPMS, specifically the US vehicle market system, is taken as the case study for investigation. Accordingly, two research tasks are accomplished. Task One is exploring potential solutions for the lack of STS dynamic data. Task Two is implementing the STS network representation and significant subnetwork mining approaches to build temporal network models and identify significant temporal subsystems.

The **priminary contribution** of this dissertation is developing a meso-level network-based framework for STS engineering and design to enrich the approaches of system science. More **specific contributions** are summarized below:

- An information retrieval and survey design framework for STS network data collection. This framework provides guidelines for two types of network data collection methods where the first one is by survey design and the second one is by web-crawling the text data and then using the NER model to extract the entity co-mentioning relationship data.
- A network motif-based STS robust design framework against seasonal effects. This framework illustrates the process of significant subsystem identification, the impacts of the obtained subsystems on the macro-level system performance, and the applicability of conducting meso-level subsystem-based STS robust design.
- A complex network-based prediction framework for STS design support with graph neural network. The proposed link predictive model within this framework outperforms the simple neural network because of the incorporation of the local network information based on GNN during prediction. In addition, this predictive model serves as an efficient tool for testing and validating design decisions.

- A micro-level entity design framework considering meso-level dependencies. To the best of our knowledge, this is the first to address the inverse problem, *i.e.*, how to achieve the desired system-level performance by promoting the formation of targeted relations among local entities.

- A comprehensive experimental framework aimed at investigating the influence of data undersampling with varying sampling ratios and model postprocessing methods on prediction outcomes using a GNN-based LP model.

- A meso-level temporal subsystem-based analyzing framework of STS dynamics. This framework shows clues for the preliminary exploration of STS dynamics based on meso-level significant temporal subsystems.

8.2 Limitations and Future Works

From the research, **four challenges and limitations** have been identified, which are summarized below:

- *Data availability.* The three advanced network techniques integrated into the proposed STS engineering and design framework, namely network motif, ERGM, and GNN, are inherently data-driven methods. For instance, both ERGM and GNN necessitate ample data for reliable model training. However, the availability of publicly accessible data for STSs has remained limited, impeding complex model training. This scarcity arises due to several reasons: Firstly, societal information in STS data is often sensitive and restricted from public dissemination. Secondly, certain STS data, such as market system data, holds significant commercial value and is not freely shared. Despite the efforts made in this dissertation to address this challenge through methods like conducting surveys, web-crawling public data, and utilizing NLP for data extraction from online sources, these approaches are constrained by scale and incur high time

costs. Consequently, the challenge of data availability persists and warrants further exploration.

- *Simplification of real-world problems by assumptions.* Despite endeavors to address real-world STS engineering and design challenges in this dissertation, certain assumptions could potentially limit the model's ability to capture reality, thereby weakening its validity. For instance, in the design case study outlined in Section 4.4, it is assumed that all STS design decisions occur at a single time point, disregarding the dynamic nature of STS design processes. Meanwhile, during the modeling of trip networks, the possibility of self-loop trips is ignored. Moreover, in Section 6.3, we formulate the optimization design problem under the assumption that the frequency of product purchases reflects the product's market share. However, it is crucial to conduct further investigation to validate this assumption. This includes reviewing existing literature and analyzing data to substantiate the rationale behind this hypothesis. Hence, continued efforts are needed to relax these assumptions.

- *Algorithm optimization for further improvement.* While this dissertation presents several networked algorithms to support STS engineering and design, certain areas require further structural or algorithmic optimization. These include: 1) In Chapter 4, the proposed adjacency matrix approximation approaches prove insufficient to validate new station installation decisions effectively. There is a need to explore alternative approximation methods to better estimate the network neighborhood information of newly introduced stations. 2) In Chapter 6, when addressing the product design optimization problem, Algorithm 1 is introduced to count the number of inter-brand triads involving the target product each time the network structure is re-predicted by ERGM. However, the computational efficiency of this algorithm is notably slow, requiring tens of hours to complete the computation. This slowdown is likely attributable to ERGM's incompatibility with GPU computing. 3) In Chapter 7, despite introducing a

comprehensive experimental framework to investigate the influence of data undersampling and model postprocessing methods on prediction outcomes using a GNN-based LP model, identifying model settings with high performance for highly imbalanced STS network data remains a challenge. Further refinement is necessary in this aspect.

- *Inadequate exploration of STS dynamics.* In Chapter 7, a preliminary exploration of STS dynamics based on meso-level significant temporal subsystems was conducted. However, further investigation is required to address two remaining challenges: 1) The first challenge concerns the dynamic data availability issue, as highlighted in the first bullet point. Section 7.2 proposed a solution by developing a GNN-based LP model. Nevertheless, as indicated in the third bullet point, the quest for a highly accurate model is ongoing. 2) The second challenge pertains to the lack of quantitative understanding of temporal network motifs. The temporal analysis in Section 7.3 culminated in an empirical examination of temporal competition motifs. Further exploration is warranted to quantitatively comprehend these patterns and integrate meso-level temporal competition information into the product design process.

To address the aforementioned challenges, several future research directions are proposed:

- *Developing advanced large language model (LLM) to extract network data from publically available text data.* Every day, a vast amount of textual content is generated by individuals in various roles, including customers, researchers, and product creators, and the majority of this text data is publicly accessible. However, unlike tabular data, which is well-formatted, unstructured text data requires multiple post-processing procedures, as illustrated in the example provided in Section 5.3. Therefore, one of our future research directions is to develop an advanced Language Understanding Model (LLM) by integrating

techniques such as Named Entity Recognition (NER) and sentiment analysis. This integration aims to extract entity interaction information from the extensive pool of public text data, thereby further addressing the data availability issues encountered in STSs.

- *Relax unrealistic assumptions to model the real-world scenarios.* Expanding upon the simplification models proposed in this dissertation, our future work aims to address the challenges posed by real-world scenarios by relaxing unrealistic assumptions. Specifically, for the design case study outlined in Section 4.4, we intend to account for self-loops and incorporate time dimension features for each design decision to capture the dynamic nature of STS design processes. Regarding the product optimization problem discussed in Chapter 6, our intention is to explore existing economic literature to ascertain the true relationship between the frequency of product purchases and market share. Additionally, we aim to utilize additional datasets, such as vehicle market data, to further investigate the validity of our proposed assumption.

- *Enhancing the accuracy of the LP model using advanced GNN techniques.* As discussed earlier, the challenge of inaccurate network prediction poses a significant obstacle in various studies within this dissertation, such as the product design framework presented in Chapter 6 and the prediction of STS dynamic network data in Chapter 7. Therefore, there is a critical need to develop a more precise LP model to enhance the reliability of these studies and their outcomes. Two potential avenues for improvement are: 1) refining the structural configuration of the GNN by leveraging recent advancements in GNN models, and 2) investigating additional undersampling and post-processing methods, and exploring their integration with GNN models to optimize performance.

- *Continued exploration of STS dynamics.* This involves two future research directions: 1) Quantifying the influence of identified temporal network motifs on both macro-level system and micro-level individual entity evolution in an STS.

This can be achieved through statistical methods (*e.g.*, counting the frequency of individual entities involved in each dynamic pattern) or neural network methods (*e.g.*, developing a dynamic GNN predictive model including these temporal patterns' information and investigating whether its inclusion improves predictive performance); 2) developing a method to integrate the information from temporal motifs into the design process of both the system and individual entities, guided by the quantification obtained.

Appendix A: Validating the linear relationship between α and β

Based on Equation 3.1, all the possible calculation of α and β, depending on whether c values are larger than 0 or not, are enumerated as follows:

1) If $c_1 > 0$, $c_2 < 0$, and $c_3 < 0$:

$$\beta = c_1, \quad \alpha = \frac{1}{2}|c_2 + c_3|, \tag{A1}$$

$$\text{So,} \quad \beta = 2\alpha. \tag{A2}$$

Similar relationship can be achieved when $c_2 > 0$, $c_1 < 0$, and $c_3 < 0$ or $c_3 > 0$, $c_1 < 0$, and $c_2 < 0$.

2) If $c_1 > 0$, $c_2 > 0$, and $c_3 < 0$:

$$\beta = \frac{1}{2}(c_1 + c_2), \quad \alpha = |c_3|, \tag{A3}$$

$$\text{So,} \quad \beta = \frac{1}{2}\alpha. \tag{A4}$$

Similar relationship can be achieved when $c_1 > 0$, $c_3 > 0$, and $c_2 < 0$ or $c_2 > 0$, $c_3 > 0$, and $c_1 < 0$.

3) If $c_1 = 0$, $c_2 > 0 (< 0)$, and $c_3 < 0 (> 0)$:

$$\beta = c_2, \quad \alpha = |c_3|, \tag{A5}$$

$$\text{So,} \quad \beta = \alpha. \tag{A6}$$

Similar relationship can be achieved when $c_1 > 0(< 0)$, $c_2 = 0$, and $c_3 < 0(> 0)$ or $c_1 > 0(< 0)$, $c_2 < 0(> 0)$, and $c_3 = 0$.

4) If $c_1 = 0$, $c_2 = 0$, and $c_3 = 0$:

$$\beta = \alpha = 0. \tag{A7}$$

Therefore, the linear relationship between α and β is validated.

Appendix B: Divvy Bike motif Z-score ranks in 2014-2016

From Table B1 to Table B3, the same trend described in Section 3.3.2 is further verified that the motifs with higher transitivity are more likely to be significant. This is also the reason that motif 238 and 46 are always ranked highest while motif 78 lowest in almost all networks.

Table B1: Divvy Bike motif Z-score ranks of each month in 2014.

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
238	238	238	46	46	46	46	46	166	46	238	238
102	174	46	238	166	166	166	166	46	102	166	102
174	46	102	102	102	102	238	102	238	166	102	46
38	102	166	166	238	238	102	238	102	238	46	166
46	38	174	38	38	38	14	164	38	38	174	174
166	166	38	174	12	164	164	14	12	14	38	38
140	140	140	140	164	14	38	12	14	12	12	140
36	12	12	12	14	12	140	38	164	164	140	12
12	36	6	14	140	140	12	140	140	140	36	36
6	6	36	164	174	174	6	174	174	174	6	6
164	164	164	6	6	6	174	6	36	6	164	164
14	14	14	36	36	36	36	36	6	36	14	14
78	78	78	78	78	78	78	78	78	78	78	78

Table B2: Divvy Bike motif Z-score ranks of each month in 2015.

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
238	238	238	238	46	46	46	46	46	46	102	238
102	174	166	166	166	166	166	166	166	166	166	102
166	102	102	46	102	102	102	102	102	102	238	46
174	46	46	102	238	238	238	238	238	238	46	166
46	166	174	38	38	38	38	38	38	38	174	174
38	38	38	174	140	140	140	140	140	174	38	38
140	140	140	140	174	12	12	12	174	140	140	140
12	12	12	12	12	174	174	174	12	12	12	12
36	36	36	164	164	164	164	164	164	14	6	6
6	6	6	14	14	14	14	14	14	6	36	36
14	164	164	36	6	6	6	6	6	164	164	164
164	14	14	6	36	36	36	36	36	36	14	14
78	78	78	78	78	78	78	78	78	78	78	78

Table B3: Divvy Bike motif Z-score ranks of each month in 2016.

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
238	238	102	102	46	46	46	46	46	46	46	238
102	102	46	46	166	166	102	102	102	102	102	102
174	166	238	166	102	102	166	166	166	166	166	166
166	46	166	238	238	238	238	238	238	238	238	174
46	174	38	38	38	38	38	38	38	38	38	46
38	38	174	174	140	12	140	174	140	174	174	38
140	140	140	140	174	140	174	140	174	140	140	140
12	12	12	12	12	174	12	12	12	12	12	12
36	36	6	6	14	14	164	6	14	36	6	36
6	6	36	164	164	164	14	14	6	14	36	6
14	164	14	36	6	6	6	164	164	6	14	14
164	14	164	14	36	36	36	36	36	164	164	164
78	78	78	78	78	78	78	78	78	78	78	78

Appendix C: Details of Optimization Problem Formulation and Solving for Extension Case

Figure C1 presents the formulated optimization problem for the extension case, where $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}, N_{R2}]$. The primary difference from the pre-update case is in the objective function, which now reflects the estimated relationship between the number of times product purchases u and the updated $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}, N_{R2}]$.

Algorithm C1 calculates the objective value for the extension case. The key differences from Algorithm 1 include: 1) In row 1, Algorithm 1, which contains all potential inter-brand triadic closures involving $P369$. In contrast, Algoritm C1 provides S_{R1} for all potential inter-brand triadic closures and S_{R2} for all potential intra-brand triadic closures involving $P369$. 2) From row 13 to row 21 in Algorithm C1, the existence probability of each intra-brand triadic closure is calculated. Correspondingly, rows 29 to 34 count the number of existing intra-brand triadic closures. 3) Row 35 in Algorithm C1 returns the objective value calculated by the updated objective function.

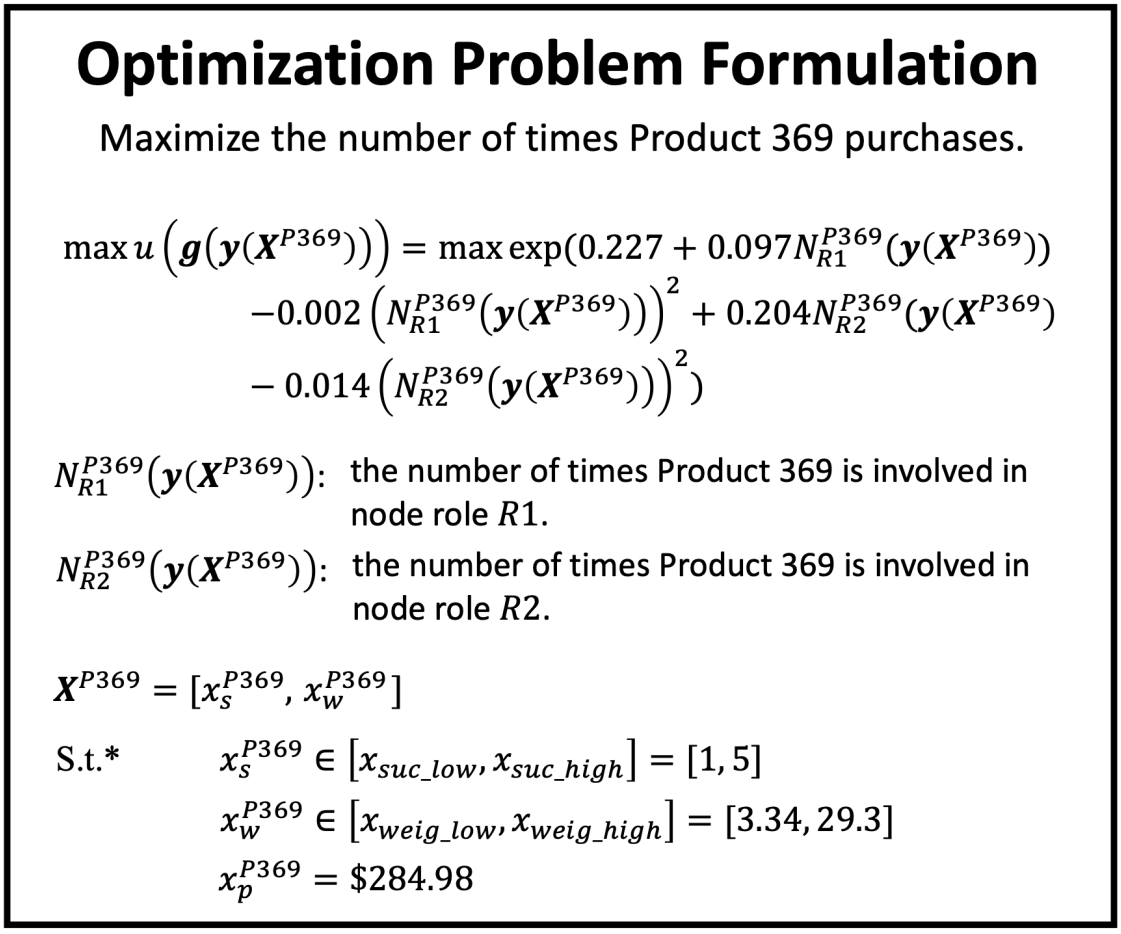


Figure C1: The optimization problem formulation corresponding to $\boldsymbol{g}(\boldsymbol{y}(\boldsymbol{X})) = [N_{R1}, N_{R2}]$.

Algorithm C1 Objective Value Calculation

Given V, S_{R1}, S_{R2}, x_s^{P369}, x_w^{P369}, x_p^{P369}, $\boldsymbol{\theta}_{est}$
Initiate $L = 100$
Simulate L networks $\boldsymbol{Y}_l, (l = 1, ..., L)$ with the given x_s^{P369}, x_w^{P369}, x_p^{P369}, and estimated ERGM parameters $\boldsymbol{\theta}_{est}$
for $m = 1$ to M_{R1} **do**
    $count = 0$
    for $l = 1$ to L **do**
        if $\boldsymbol{y}^m_{P369,V_i,V_j}$ exists in $\boldsymbol{Y}_l$ **then**
            $count = count + 1$
        end if
    end for
    $Pr(\boldsymbol{y}^m_{P369,V_i,V_j}) = count/L$
end for
for $m = 1$ to M_{R2} **do**
    $count = 0$
    for $l = 1$ to L **do**
        if $\boldsymbol{y}^m_{P369,V_i,V_j}$ exists in $\boldsymbol{Y}_l$ **then**
            $count = count + 1$
        end if
    end for
    $Pr(\boldsymbol{y}^m_{P369,V_i,V_j}) = count/L$
end for
$Pr_{threshold} = \text{Median}(Pr(\boldsymbol{y}^m_{P369,V_i,V_j}))$
for $m = 1$ to M_{R1} **do**
    $N_{R1}^{P369} = 0$
    if $Pr(\boldsymbol{y}^m_{P369,V_i,V_j}) > Pr_{threshold}$ **then**
        $N_{R1}^{P369} = N_{R1}^{P369} + 1$
    end if
end for
for $m = 1$ to M_{R2} **do**
    $N_{R2}^{P369} = 0$
    if $Pr(\boldsymbol{y}^m_{P369,V_i,V_j}) > Pr_{threshold}$ **then**
        $N_{R2}^{P369} = N_{R2}^{P369} + 1$
    end if
end for
Return $u = exp(0.227 + 0.097N_{R1}^{P369} - 0.002(N_{R1}^{P369})^2) + 0.204N_{R2}^{P369} - 0.014(N_{R2}^{P369})^2)$

Bibliography

Vacuum cleaner specifications. https://www.bestvacuum.com/pages/vacuum-cleaner-specifications. Accessed: 01/01/2024.

Faez Ahmed, Yaxin Cui, Yan Fu, and Wei Chen. A graph neural network approach for product relationship prediction. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 85383, page V03AT03A036. American Society of Mechanical Engineers, 2021.

Faez Ahmed, Yaxin Cui, Yan Fu, and Wei Chen. Product Competition Prediction in Engineering Design Using Graph Neural Networks. *ASME Open Journal of Engineering*, 1, 05 2022a. ISSN 2770-3495. doi: 10.1115/1.4054299. URL https://doi.org/10.1115/1.4054299. 011020.

Faez Ahmed, Yaxin Cui, Yan Fu, and Wei Chen. Product competition prediction in engineering design using graph neural networks. *ASME Open Journal of Engineering*, 1:011020, 2022b.

Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

J Anon. Announcement: Reducing our irreproducibility. *Nature*, 496(7446): 398, 2013.

Sinan Aral and Dylan Walker. Creating social contagion through viral product design: A randomized trial of peer influence in networks. *Management science*, 57(9):1623–1639, 2011.

Jennifer J Argo. A contemporary review of three types of social influence in consumer psychology. *Consumer Psychology Review*, 3(1):126–140, 2020.

Jasbir Singh Arora. *Introduction to optimum design*. Elsevier, 2004.

Ruibin Bai, Stein W Wallace, Jingpeng Li, and Alain Yee-Loong Chong. Stochastic service network design with rerouting. *Transportation Research Part B: Methodological*, 60:50–65, 2014.

Benjamin Baiser, Rasha Elhesha, and Tamer Kahveci. Motifs in the assembly of food web networks. *Oikos*, 125(4):480–491, 2016.

Qifang Bao, Ekaterina Sinitskaya, Kelley J Gomez, Erin F MacDonald, and Maria C Yang. A human-centered design approach to evaluating factors in residential solar pv adoption: A survey of homeowners in california and massachusetts. *Renewable energy*, 151:503–513, 2020.

Albert-László Barabási. The network takeover. *Nature Physics*, 8(1):14–16, 2012.

Albert-László Barabási and MÃ¡rton PÃ3sfai. *Network Science*. Cambridge University Press, 2016. ISBN 9781107076266. URL `https://books.google.com/books?id=iLtGDQAAQBAJ`.

Amanda S Barnard, Jordan J Louviere, Edward Wei, and Leon Zadorin. Using hypothetical product configurators to measure consumer preferences for nanoparticle size and concentration in sunscreens. *Design Science*, 2:e12, 2016.

Gordon Baxter and Ian Sommerville. Socio-technical systems: From design methods to systems engineering. *Interacting with computers*, 23(1):4–17, 2011.

Lauren C Beaumont, Lucy E Bolton, Alison McKay, and Helen PN Hughes. Rethinking service design: a socio-technical approach to the development of business models. *Product Development in the Socio-sphere: Game Changing Paradigms for 21st Century Breakthrough Product Development and Innovation*, pages 121–141, 2014.

Yoshua Bengio and Yves Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Advances in Neural Information Processing Systems*, 16, 2003.

Youyi Bi, Jian Xie, Zhenghui Sha, Mingxian Wang, Yan Fu, and Wei Chen. Modeling spatiotemporal heterogeneity of customer preferences in engineering design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 51753, page V02AT03A050. American Society of Mechanical Engineers, 2018.

Youyi Bi, Yunjian Qiu, Zhenghui Sha, Mingxian Wang, Yan Fu, Noshir Contractor, and Wei Chen. Modeling multi-year customers' considerations and choices in china's auto market using two-stage bipartite network analysis. *Networks and Spatial Economics*, 21:365–385, 2021.

Per Block. Reciprocity, transitivity, and the mysterious three-cycle. *Social Networks*, 40:163–173, 2015.

Hamparsum Bozdogan. Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52(3): 345–370, 1987.

Jason Brownlee. *Imbalanced classification with python: Better metrics, balance skewed classes, cost-sensitive learning.* Machine Learning Mastery, 2020.

M Cecilia Bustamante and Andres Donangelo. Product market competition and industry returns. *The Review of Financial Studies*, 30(12):4216–4266, 2017.

Carter T Butts, Martina Morris, Pavel N Krivitsky, Zack Almquist, Mark S Handcock, David R Hunter, Steven M Goodreau, and Skye Bender de Moll. Introduction to exponential-family random graph (erg or p*) modeling with

ergm. *European University Institute, Florence. URL: http://cran. r-project. org/web/packages/ergm/vignettes/ergm. pdf*, 2014.

Karen E Campbell and Barrett A Lee. Name generators in surveys of personal networks. *Social networks*, 13(3):203–221, 1991.

Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

Haochen Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. A tutorial on network embeddings. *arXiv preprint arXiv:1808.02590*, 2018.

Sarvenaz Choobdar, Pedro Ribeiro, and Fernando Silva. Motif mining in weighted networks. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 210–217. IEEE, 2012. DOI: 10.1109/ICDMW.2012.111.

Valentino Crespi, Aram Galstyan, and Kristina Lerman. Top-down vs bottom-up methodologies in multi-agent system design. *Autonomous Robots*, 24(3): 303–313, 2008.

Yaxin Cui, Faez Ahmed, Zhenghui Sha, Lijun Wang, Yan Fu, and Wei Chen. A weighted network modeling approach for analyzing product competition. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 84003, page V11AT11A036. American Society of Mechanical Engineers, 2020. DOI: 10.1115/DETC2020-22591.

Anderson Andrei De Bona, Marcelo de Oliveira Rosa, Keiko Verônica Ono Fonseca, and Ricardo Lüders. A reduced model for complex network analysis of public transportation systems. *Physica A: Statistical Mechanics and its Applications*, 567:125715, 2021.

Divvy_Bike. Divvy system data, 2020. URL `https://www.divvybikes.com/system-data`. Last accessed 21 February 2020.

Alan F Dutka and Howard H Hansen. *Fundamentals of data normalization*. Addison-Wesley Longman Publishing Co., Inc., 1991.

Jon D Elhai, Patrick S Calhoun, and Julian D Ford. Statistical procedures for analyzing mental health services data. *Psychiatry research*, 160(2):129–136, 2008.

Waguih ElMaraghy, Hoda ElMaraghy, Tetsuo Tomiyama, and Laszlo Monostori. Complexity in engineering design and manufacturing. *CIRP annals*, 61 (2):793–814, 2012. DOI: 10.1016/j.cirp.2012.05.001.

Diane Felmlee, Cassie McMillan, Don Towsley, and R Whitaker. Social network motifs: A comparison of building blocks across multiple social networks. In *Annual Meetings of the American Sociological Association, Philadelphia, US*, 2018.

Vincenzo Ferrero, Bryony DuPont, Kaveh Hassani, and Daniele Grandi. Classifying component function in product assemblies with graph neural networks. *Journal of Mechanical Design*, 144(2), 2022.

Øystein D Fjeldstad, Julie K Johnson, Peter A Margolis, Michael Seid, Pär Höglund, and Paul B Batalden. Networked health care: rethinking value creation in learning health care systems. Technical report, Wiley Online Library, 2020.

Robin Flowerdew and David Martin. *Methods in human geography: a guide for students doing a research project*. Pearson Education, 2005.

J Sophia Fu, Zhenghui Sha, Yun Huang, Mingxian Wang, Yan Fu, and Wei Chen. Two-stage modeling of customer choice preferences in engineering design using bipartite network analysis. In *International Design Engineering*

Technical Conferences and Computers and Information in Engineering Conference, volume 58127, page V02AT03A039. American Society of Mechanical Engineers, 2017. DOI: 10.1115/DETC2017-68099.

Paolo Gabrielli, Florian Fürer, Georgios Mavromatidis, and Marco Mazzotti. Robust and optimal design of multi-energy systems with seasonal storage through uncertainty analysis. *Applied energy*, 238:1192–1210, 2019.

Salvador García, Julián Luengo, and Francisco Herrera. *Data preprocessing in data mining*, volume 72. Springer, 2015.

Phillip AO Gavino, Yinshuang Xiao, Yaxin Cui, Wei Chen, and Zhenghui Sha. Evolutionary co-mention network analysis via social media mining. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 87301, page V03AT03A045. American Society of Mechanical Engineers, 2023.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Steven D Gribble. Robustness in complex systems. In *Proceedings eighth workshop on hot topics in operating systems*, pages 21–26. IEEE, 2001. DOI: 10.1109/HOTOS.2001.990056.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

Suining He and Kang G Shin. Towards fine-grained flow forecasting: A graph attention approach for bike sharing systems. In *Proceedings of The Web Conference 2020*, pages 88–98, 2020.

Babak Heydari, Zoe Szajnfarber, Jitesh Panchal, Michel-Alexandre Cardin, Katja Holtta-Otto, and Gül E. Kremer. Special Issue: Analysis and Design of Sociotechnical Systems. *Journal of Mechanical Design*, 142(12), 11 2020. ISSN 1050-0472. doi: 10.1115/1.4048699. URL https://doi.org/10.1115/1.4048699. 121101.

Paul W Holland and Samuel Leinhardt. The statistical analysis of local structure in social networks. *National Bureau of Economic Research, Inc, NBER Working Papers*, 6, 1974. DOI: 10.3386/w0044.

David R Hunter. Curved exponential family models for social networks. *Social networks*, 29(2):216–230, 2007.

Monu Kalsi, Kurt Hacker, and Kemper Lewis. A comprehensive robust design approach for decision trade-offs in complex systems design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 19715, pages 1343–1354. American Society of Mechanical Engineers, 1999. DOI: 10.1115/1.1334596.

Christo Karuna. Industry product market competition and managerial incentives. *Journal of accounting and economics*, 43(2-3):275–297, 2007.

Nadav Kashtan, S Itzkovitz, R Milo, and U Alon. Mfinder tool guide. *Department of Molecular Cell Biology and Computer Science and Applied Math., Weizmann Inst. of Science, Rehovot Israel, technical report*, 2002.

Nadav Kashtan, Shalev Itzkovitz, Ron Milo, and Uri Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004. DOI: 10.1093/bioinformatics/bth163.

James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

Ali Keyhani. *Design of smart power grid renewable energy systems.* John Wiley & Sons, 2016.

Philip Kotler, Kevin L Keller, Fabio Ancarani, and Michele Costabile. *Marketing management 14/e.* Pearson, 2014.

Mitch Kramer. Best practices in systems development lifecycle: An analyses based on the waterfall model. *Review of Business & Finance Studies*, 9(1): 77–84, 2018.

Pavel N. Krivitsky, David R. Hunter, Martina Morris, and Chad Klumb. ergm 4: New features for analyzing exponential-family random graph models. *Journal of Statistical Software*, 105(6):1–44, 2023. doi: 10.18637/jss.v105.i06. URL `https://www.jstatsoft.org/index.php/jss/article/view/v105i06`.

Thomas Y Lee and Eric T Bradlow. Automated marketing research using online customer reviews. *Journal of Marketing Research*, 48(5):881–894, 2011.

Ai Qiang Li, Nicholas Rich, Pauline Found, Maneesh Kumar, and Steve Brown. Exploring product–service systems in the digital era: a socio-technical systems perspective. *The TQM Journal*, 32(4):897–913, 2020.

Junming Liu, Leilei Sun, Qiao Li, Jingci Ming, Yanchi Liu, and Hui Xiong. Functional zone based hierarchical demand prediction for bike system expansion. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 957–966, 2017.

Shmoolik Mangan and Uri Alon. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21): 11980–11985, 2003.

Liliane Manny. Socio-technical challenges towards data-driven and integrated urban water management: A socio-technical network approach. *Sustainable Cities and Society*, 90:104360, 2023.

Liliane Manny, Mario Angst, Jörg Rieckermann, and Manuel Fischer. Socio-technical networks of infrastructure management: Network concepts and motifs for studying digitalization, decentralization, and integrated management. *Journal of environmental management*, 318:115596, 2022.

Samuel A Markolf, Christopher Hoehne, Andrew Fraser, Mikhail V Chester, and B Shane Underwood. Transportation resilience to climate change and extreme weather events–beyond risk and robustness. *Transport policy*, 74:174–186, 2019. DOI: 10.1016/j.tranpol.2018.11.003.

Joaquim RRA Martins and Andrew Ning. *Engineering design optimization*. Cambridge University Press, 2021.

MATLAB. Matlab genetic algorithm, 2020. URL `https://www.mathworks.com/help/gads/ga.html#d122e41247`. Last accessed 21 February 2020.

Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 2012.

Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

Martina Morris, Mark S Handcock, and David R Hunter. Specification of exponential-family random graph models: terms and computational aspects. *Journal of statistical software*, 24(4):1548, 2008.

Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA, 2015.

Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python, 2014–. URL `https://github.com/fmfn/BayesianOptimization`.

Giuliano Andrea Pagani and Marco Aiello. The power grid as a complex network: a survey. *Physica A: Statistical Mechanics and its Applications*, 392 (11):2688–2700, 2013.

A Paranjape, AR Benson, and J Leskovec. Motifs in temporal networks in: Proceedings of the international conference on web search and data mining, 601–610, 2017. DOI: 10.1145/3018661.3018731.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

Christian Pescher and Martin Spann. Relevance of actors in bridging positions for product-related information diffusion. *Journal of Business Research*, 67(8): 1630–1637, 2014.

Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. Semantic cosine similarity. In *The 7th international student conference on advanced science and technology ICAST*, volume 4, page 1, 2012.

Rahul Rai. Identifying key product attributes and their importance levels from online customer reviews. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 45028, pages 533–540. American Society of Mechanical Engineers, 2012.

Toqir A Rana and Yu-N Cheah. A two-fold rule-based model for aspect extraction. *Expert systems with applications*, 89:273–285, 2017.

F Rasche and S Wernicke. Fanmod fast network motif detection–manual. *Bioinformatics*, 22(9):1152–1153, 2006.

Charles Rathkopf. Network representation and complex systems. *Synthese*, 195(1):55–78, 2018.

Pedro Ribeiro, Pedro Paredes, Miguel EP Silva, David Aparicio, and Fernando Silva. A survey on subgraph counting: concepts, algorithms, and applications to network motifs and graphlets. *ACM Computing Surveys (CSUR)*, 54(2): 1–36, 2021.

Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p*) models for social networks. *Social networks*, 29(2):173–191, 2007.

Luis A Rodriguez, Abheek Chatterjee, and Astrid Layton. Ecological decentralization for improving the resilient design of urban water distribution networks. In *Conference on Systems Engineering Research*, pages 587–601. Springer, 2023.

Jose Antonio Rosa, Joseph F Porac, Jelena Runser-Spanjol, and Michael S Saxon. Sociocognitive dynamics in a product market. *Journal of marketing*, 63(4_suppl1):64–77, 1999.

Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, Sungchul Kim, Anup Rao, and Yasin Abbasi Yadkori. Hone: Higher-order network embeddings. *arXiv preprint arXiv:1801.09303*, 2018.

Ron Sanchez. Strategic flexibility in product competition. *Strategic management journal*, 16(S1):135–159, 1995.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Henning Schwöbbermeyer. *Network Motifs*, chapter 5, pages 85–111. John Wiley & Sons, Ltd, 2008. ISBN 9780470253489. doi: https://doi.org/10.1002/9780470253489.ch5. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470253489.ch5`.

Zhenghui Sha. *Decision-Centric Foundations for Complex Systems Engineering and Design*. PhD thesis, Purdue University, 2015.

Zhenghui Sha and Jitesh H Panchal. Estimating the node-level behaviors in complex networks from structural datasets. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 55867, page V02BT02A003. American Society of Mechanical Engineers, 2013a. DOI: 10.1115/DETC2013-12063.

Zhenghui Sha and Jitesh H Panchal. Towards the design of complex evolving networks with high robustness and resilience. *Procedia Computer Science*, 16: 522–531, 2013b. DOI:10.1016/j.procs.2013.01.055.

Zhenghui Sha and Jitesh H Panchal. A degree-based decision-centric model for complex networked systems. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 50084, page V01BT02A016. American Society of Mechanical Engineers, 2016. DOI:10.1115/DETC2016-60036.

Zhenghui Sha, Veronica Saeger, Mingxian Wang, Yan Fu, and Wei Chen. Analyzing customer preference to product optional features in supporting product configuration. *SAE International Journal of Materials and Manufacturing*, 10 (3):320–332, 2017.

Zhenghui Sha, Yun Huang, Jiawei Sophia Fu, Mingxian Wang, Yan Fu, Noshir Contractor, and Wei Chen. A network-based approach to modeling and predicting product coconsideration relations. *Complexity*, 2018, 2018. DOI: 10.1155/2018/2753638.

Zhenghui Sha, Youyi Bi, Mingxian Wang, Amanda Stathopoulos, Noshir Contractor, Yan Fu, and Wei Chen. Comparing utility-based and network-based

approaches in modeling customer preferences for engineering design. In *Proceedings of the Design Society: International Conference on Engineering Design*, volume 1, pages 3831–3840. Cambridge University Press, 2019a. DOI: 10.1017/dsi.2019.390.

Zhenghui Sha, Ashish M Chaudhari, and Jitesh H Panchal. Modeling participation behaviors in design crowdsourcing using a bipartite network-based approach. *Journal of Computing and Information Science in Engineering*, 19 (3), 2019b. DOI: 10.1115/1.4042639.

Zhenghui Sha, Yaxin Cui, Yinshuang Xiao, Amanda Stathopoulos, Noshir Contractor, Yan Fu, and Wei Chen. A network-based discrete choice model for decision-based design. *Design Science*, 9:e7, 2023.

Ping Shao, Yang Yang, Shengyao Xu, and Chunping Wang. Network embedding via motifs. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(3):1–20, 2021.

Ahmed Zayed AL Shaqsi, Kamaruzzaman Sopian, and Amer Al-Hinai. Review of energy storage services, applications, limitations, and benefits. *Energy reports*, 6:288–306, 2020.

Allan D Shocker, Moshe Ben-Akiva, Bruno Boccara, and Prakash Nedungadi. Consideration set influences on consumer decision-making and choice: Issues, models, and suggestions. *Marketing letters*, 2:181–197, 1991.

Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *iEEE Access*, 9:79143–79168, 2021.

Harold Soh, Sonja Lim, Tianyou Zhang, Xiuju Fu, Gary Kee Khoon Lee, Terence Gih Guang Hung, Pan Di, Silvester Prakasam, and Limsoon Wong. Weighted complex network analysis of travel routes on the singapore public

transportation system. *Physica A: Statistical Mechanics and its Applications*, 389(24):5852–5863, 2010.

B Song, C McComb, and F Ahmed. Assessing machine learnability of image and graph representations for drone performance prediction. *Proceedings of the Design Society*, 2:1777–1786, 2022.

Ran Spiegler. Choice complexity and market competition. *Annual Review of Economics*, 8:1–25, 2016.

Zdenko Stanicek and Marek Winkler. Service systems through the prism of conceptual modeling. *Service Science*, 2(1-2):112–125, 2010.

Alina Stankevich. Explaining the consumer decision-making process: Critical literature review. *Journal of international business research and marketing*, 2 (6), 2017.

Lewi Stone, Daniel Simberloff, and Yael Artzy-Randrup. Network motifs and their origins. *PLoS computational biology*, 15(4):e1006749, 2019. DOI: 10.1371/journal.pcbi.1006749.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine De Marneffe, and Wei Xu. Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, 2016.

Xiaoqian Sun, Sebastian Wandelt, and Florian Linke. Temporal evolution analysis of the european air transportation system: air navigation route network and airport network. *Transportmetrica B: Transport Dynamics*, 3(2):153–168, 2015. DOI: 10.1080/21680566.2014.960504.

Shun Takai, Vivek K Jikar, and Kenneth M Ragsdell. An approach toward integrating top-down and bottom-up product concept and design selection. 2011.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

Ke Tao, Fabian Abel, Claudia Hauff, Geert-Jan Houben, and Ujwal Gadiraju. Groundhog day: near-duplicate detection on twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1273–1284, 2013.

Khoinguyen Trinh and Zhenghui Sha. Toward the design of artificial swarms using network motifs. In *Recent Trends and Advances in Model Based Systems Engineering*, pages 603–618. Springer, 2022.

Eric Lansdown Trist and Kenneth W Bamforth. Some social and psychological consequences of the longwall method of coal-getting: An examination of the psychological situation and defences of a work group in relation to the social structure and technological content of the work system. *Human relations*, 4(1): 3–38, 1951. DOI:10.1177/001872675100400101.

Suppawong Tuarob and Conrad S. Tucker. Quantifying Product Favorability and Extracting Notable Product Features Using Large Scale Social Media Data. *Journal of Computing and Information Science in Engineering*, 15(3):031003, 09 2015. ISSN 1530-9827. doi: 10.1115/1.4029562. URL `https://doi.org/10.1115/1.4029562`.

Paul T Von Hippel. Mean, median, and skew: Correcting a textbook rule. *Journal of statistics Education*, 13(2), 2005.

Junwei Wang. *Towards a resilient networked service system.* PhD thesis, University of Saskatchewan, 2013.

Mingxian Wang, Wei Chen, Yan Fu, and Yong Yang. Analyzing and predicting heterogeneous customer preferences in china's auto market using choice

modeling and network analysis. *SAE International Journal of Materials and Manufacturing*, 8(3):668–677, 2015.

Mingxian Wang, Wei Chen, Yun Huang, Noshir S Contractor, and Yan Fu. Modeling customer preferences using multidimensional network analysis in engineering design. *Design Science*, 2, 2016a. DOI: https://doi.org/10.1017/dsj.2016.11.

Mingxian Wang, Zhenghui Sha, Yun Huang, Noshir Contractor, Yan Fu, and Wei Chen. Forecasting technological impacts on customers' co-consideration behaviors: a data-driven network analysis approach. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 50107, page V02AT03A040. American Society of Mechanical Engineers, 2016b. DOI: 10.1115/DETC2016-60015.

Mingxian Wang, Zhenghui Sha, Yun Huang, Noshir Contractor, Yan Fu, and Wei Chen. Predicting product co-consideration and market competitions for technology-driven product design: a network-based approach. *Design Science*, 4, 2018. DOI: 10.1017/dsj.2018.4.

Z. Wang, S. Azarm, and P. K. Kannan. Strategic Design Decisions for Uncertain Market Systems Using an Agent Based Approach. *Journal of Mechanical Design*, 133(4):041003, 05 2011. ISSN 1050-0472. doi: 10.1115/1.4003843. URL `https://doi.org/10.1115/1.4003843`.

Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*. Cambridge university press, 1994.

Sebastian Wernicke and Florian Rasche. Fanmod: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152–1153, 2006.

Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4: 65–85, 1994.

OpenStreetMap Wiki. Overpass turbo — openstreetmap wiki,, 2022. URL https://wiki.openstreetmap.org/w/index.php?title=Overpass_turbo&oldid=2247595. [Online; accessed 4-February-2022].

Haixia Wu, Chunyao Song, Yao Ge, and Tingjian Ge. Link prediction on complex networks: an experimental survey. *Data Science and Engineering*, 7 (3):253–278, 2022.

Jiaxin Wu and Pingfeng Wang. Generative design for resilience of interdependent network systems. *Journal of Mechanical Design*, 145(3):031705, 2023.

Y Xiao and Y Cui. Product competition analysis for engineering design: A network mining approach. In *2023 Conference on Systems Engineering Research (CSER)*, 2023.

Yinshuang Xiao and Zhenghui Sha. Towards engineering complex socio-technical systems using network motifs: A case study on bike-sharing systems. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 84003, page V11AT11A045. American Society of Mechanical Engineers, 2020. DOI: 10.1115/DETC2020-22631.

Yinshuang Xiao and Zhenghui Sha. Robust design of complex socio-technical systems against seasonal effects: a network motif-based approach. *Design Science*, 8, 2022.

Yinshuang Xiao, Faez Ahmed, and Zhenghui Sha. Travel links prediction in shared mobility networks using graph neural network models. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 86212, page V002T02A079. American Society of Mechanical Engineers, 2022a.

Yinshuang Xiao, Yaxin Cui, Nikita Raut, Jonathan Haris Januar, Johan Koskinen, Noshir Contractor, Wei Chen, and Zhenghui Sha. Information retrieval

and survey design for two-stage customer preference modeling. *Proceedings of the Design Society*, 2:811–820, 2022b.

Yinshuang Xiao, Faez Ahmed, and Zhenghui Sha. Graph neural network-based design decision support for shared mobility systems. *Journal of Mechanical Design*, 145(9), 2023a.

Yinshuang Xiao, Yaxin Cui, Michael T Cardone, Wei Chen, and Zhenghui Sha. Product competition analysis for engineering design: A network mining approach. *2023 Conference on Systems Engineering Research (CSER)*, 2023b. URL `https://par.nsf.gov/biblio/10416558`.

Yinshuang Xiao, Yaxin Cui, Nikita Raut, Jonathan Januar, Johan Koskinen, Noshir Contractor, Wei Chen, and Zhenghui Sha. Survey data on customer two-stage decision-making process in household vacuum cleaner market. *Data in Brief*, 54:110353, 2024. ISSN 2352-3409. doi: https://doi.org/10.1016/j.dib.2024.110353. URL `https://www.sciencedirect.com/science/article/pii/S2352340924003226`.

Banghua Xie, Xiaoge Tian, Liulin Kong, and Weiming Chen. The vulnerability of the power grid structure: A system analysis based on complex network theory. *Sensors*, 21(21):7097, 2021.

Jian Xie, Youyi Bi, Zhenghui Sha, Mingxian Wang, Yan Fu, Noshir Contractor, Lin Gong, and Wei Chen. Data-driven dynamic network modeling for analyzing the evolution of product competitions. *Journal of Mechanical Design*, 142(3): 031112, 2020.

Yong Yang and Ana V Diez-Roux. Walking distance by trip purpose and population subgroups. *American journal of preventive medicine*, 43(1):11–19, 2012.

Xiao Yinshuang, Ahmed Faez, and Zhenghui Sha. Travel links prediction in shared mobility networks using graph neural network models. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2022.

Arthur HC Yip, Jeremy J Michalek, and Kate S Whitefoot. Implications of competitor representation for profit-maximizing design. *Journal of Mechanical Design*, 144(1):011705, 2022.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Vita

Yinshuang Xiao

Email: yinshuangxiao@utexas.edu

Tel: +1 479-301-0254

Personal Website: https://xiaoyinshuang.github.io/

Education:

- **Ph.D. in Mechanical Engineering** — Expected May 2024
  The University of Texas at Austin — *Austin, TX, USA*
- **Master of Science in Mechanical Engineering** — Jun. 2018
  University of Electronic Science and Technology of China — *Chengdu, China*
- **Bachelor of Science in Mechanical Engineering** — Jun. 2014
  University of Electronic Science and Technology of China — *Chengdu, China*

Professional Development:

- **Data Scientist Intern** — Jun. 2023 - Jan. 2024
  Ford Motor Company — *Remote*
- **Graduate Research Assistant** — Aug. 2021 - Present
  The University of Texas at Austin — *Austin, TX, USA*
- **Graduate Research Assistant** — Aug. 2019 - Jul. 2021
  The University of Arkansas — *Fayetteville, AR, USA*
- **New Energy Vehicles R&D Engineer** — Aug. 2018 - Aug. 2019
  Shanghai Volkswagen Automotive — *Shanghai, China*

Publication:

Journal Articles

[1] **Y. Xiao**, Y. Cui, W. Chen, J. Koskinen, N. Contractor, Z. Sha, "Network-Based Complex System Engineering Optimization Design With Considering Local Dependencies," *Journal of Mechanical Design. In Review.*

[2] Y. Cui, Z. Sun, **Y. Xiao**, Z. Sha, J. Koskinen, N. Contractor, W. Chen, "Network-Based Analysis of Heterogeneous Customer Preferences in Consideration-then-Choice Decision-Making with Market Segmentation," *Journal of Computing and Information Science in Engineering. Accepted.*

[3] **Y. Xiao**, Y. Cui, N. Raut, J. Januar, J. Koskinen, N. Contractor, W. Chen, Z. Sha, "Survey Data on Customer Two-Stage Decision-Making Process in Household Vacuum Cleaner Market," *Data in Brief*, 54, p.110353, 2024.

[4] **Y. Xiao**, F. Ahmed, Z. Sha, "Graph Neural network-based design decision support for shared mobility systems," *Journal of Mechanical Design*, volume 145, issue 9, pp: 091703 (13), 2023.

[5] Z. Sha, Y. Cui, **Y. Xiao**, A. B. Stathopoulos, N. Contractor, Y. Fu, W. Chen, "A Network-Based Discrete Choice Model for Decision-Based Design," *Design Science*, 9, E7, 2023.

[6] **Y. Xiao**, Z. Sha, "Robust Design of Complex Socio-Technical Systems against Seasonal Effects: A Network Motif-Based Approach," *Design Science*, 8, E2, 2022.

[7] **Y. Xiao**, D. Ren, P. Xiao, P. Du, "An Equivalent Modeling Method for the Radiated Electromagnetic Interference of PCB Based on Near-field Scanning," *Applied Computational Electromagnetics Society Journal*, 34(5), 2019.

Refereed Conference Papers

[8] **Y. Xiao**, Z. Sha, "Graph Neural Network-Based Link Prediction for Highly Imbalanced Network Data," *ASME 2024 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Washington, DC, USA, Aug 25-28, 2024. *Accepted.*

[9] B. Thongmak, **Y. Xiao**, A. Layton, Z. Sha, "From Plant-Pollinator to Product-Customer: Bio-Inspired Network Modularity Analysis in Design for Market Systems," *The 21st Annual Conference on Systems Engineering Research (CSER 2024)*, Tucson, Arizona, Mar 25-27, 2024.

[10] B. Thongmak, **Y. Xiao**, P. Gavino, M. Zhang, Z. Sha, "Geospatial Network Analysis of US Megaregions in 40 Years," *The 57th Hawaii International Conference on System Science (HICSS)*, Maui, HI, Jan. 3-6, 2024.

[11] P. Gavino, **Y. Xiao**, Y. Cui, W. Chen, Z. Sha, "Evolutionary Co-Mention Network Analysis via Social Media Mining," *ASME 2023 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Boston, MA, Aug. 20-23, 2023.

[12] **Y. Xiao**, Y. Cui, M. Cardone, W. Chen, Z. Sha, "Product Competition Analysis for Engineering Design: A Network Mining Approach," *The 20th Annual Conference on Systems Engineering Research (CSER 2023)*, Hoboken, New Jersey, Mar 16-17, 2023.

[13] Y. Cui, **Y. Xiao**, Z. Sha, W. Chen, "Network-Based Analysis of Heterogeneous Consideration-then Choice Customer Preferences with Market Segmentations," *The 20th Annual Conference on Systems Engineering Research (CSER 2023)*, Hoboken, New Jersey, Mar 16-17, 2023.

[14] **Y. Xiao**, F. Ahmed, Z. Sha, "Travel Links Prediction In Shared Mobility Networks Using Graph Neural Network Models," *ASME 2022 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, St. Louis, Missouri, Aug. 14-17, 2022.

[15] **Y. Xiao**, Y. Cui, N. Raut, J. H. Januar, J. Koskinen, N. Contractor, W. Chen, Z. Sha, "Information Retrieval and Survey Design For Two-Stage Customer Preference Modeling," *The 17th International Design Conference*, Cavtat, Croatia, May 23-26, 2022.

[16] **Y. Xiao**, Z. Sha, "Towards Engineering Complex Sociotechnical Systems Using Network Motifs: A Case Study on Bike-Sharing Systems," *ASME 2020 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Virtual, Online, Aug 17-19, 2020.

Conference Abstracts and Posters

[17] M. Zhang, B. Thongmak, **Y. Xiao**, P. Gavino, Z. Sha, L. Zhao "Explore U.S. Megaregion Dynamics from a Network Science Perspective," *The 1st International Conference on Urban Science and Sustainability*, Xiamen, China, Dec. 14-18, 2023. *Accepted*.

[18] **Y. Xiao**, Y. Cui, W. Chen, N. Contractor, J. Koskinen, Z. Sha, "Design for Market Systems with Network-Based Product Competition Analysis," *9th International Engineering Systems Symposium: CESUN 2023*, Evanston, Illinois, Nov 6-7, 2023.

[19] **Y. Xiao**, Z. Sha, "Socio-Technical Systems Engineering and Design: A Meso-Level Network-Based Approach," DTM Student Poster Competition, *ASME 2022 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, St. Louis, Missouri, Aug. 14-17, 2022. **(Won the Trave Award).**

[20] **Y. Xiao**, Y. Cui, W. Chen, J. Koskinen, N. Contractor, Z. Sha, "A Network-Based Approach to Modeling Product Co-consideration and Choice Relations," *Sunbelt 2022 – The XLII International Sunbelt Social Networks Conference*, Cairns, Australia, Jul 12-16, 2022.

[21] Y. Cui, **Y. Xiao**, Z. Sha, N. Contractor, J. Koskinen, W. Chen, "Network-based Customer Preference Modeling," *Sunbelt 2022 – The XLII International Sunbelt Social Networks Conference*, Cairns, Australia, Jul 12-16, 2022.

[22] **Y. Xiao**, Z. Sha, "Robust Design of Complex Socio-Technical Systems using Complex Networks," CIE Graduate Research Poster Competition, *ASME 2021*

International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Virtual, Online, Aug 17-19, 2021. **(Won the Trave Award).**

[23] **Y. Xiao**, Z. Sha, "A Network Motifs-Based Approach to Improving Robustness of Complex Socio-Technical Systems Against Seasonal Effects," *Networks 2021: A Joint Sunbelt and NetSci Conference*, Virtual, Online, Jul. 6-11, 2021. Extended Abstract and Oral Presentation.

MS Thesis

Y. Xiao, "An Equivalent Modeling Method for the Electromagnetic Radiation of PCB Based on Near-Field Scanning," presented to the faculty of The School of Mechanical and Electrical Engineering, June 2018, University of Electronic Science and Technology of China, Sichuan, China.

Awards:

ASME IDETC-CIE, Student Hackathon — First Place Aug. 2022

- Awarded by ASME Computer and Information in Engineering Division,
- Award amount: $1300.

ASME IDETC-CIE, DTM PhD Student Poster Session — Travel Award Aug. 2022

- Awarded by ASME Design Engineering Division,
- Award amount: $1000 (Only the **top ten** abstracts were selected).

ASME IDETC-CIE, 2021 Graduate Research Poster Session — Travel Award Aug. 2021

- Awarded by ASME Computer and Information in Engineering Division,
- Award amount: $200.

ASME IDETC-CIE, Student Hackathon — Third Place Aug. 2021

- Awarded by ASME Computer and Information in Engineering Division,

- Award amount: $500.

ASME IDETC-CIE, Student Hackathon — Third Place Nov. 2020
- Awarded by ASME Computer and Information in Engineering Division,
- Award amount: $500.

Teaching and Mentoring:

Guest Lecturer Fall. 2022

- Course: ME 397 Data-Driven Design And Decision-Making In Complex Systems (Walker Department of Mechanical Engineering, UT Austin)
- Conducted engaging guest lectures on deep learning applications within the realm of complex socio-technical system engineering and design for ME 397.
- Developed supplemental materials and resources to enhance student understanding.
- Received positive feedback from students for clarity and effectiveness of presentations.

Undergraduate Mentor Jun. 2021 - Aug. 2023

- Project: A Hierarchical Multidimensional Network-based Approach for Multi-Competitor Product Design
  (Collaborative Project Between UT Austin & Northwestern)
- Mentored undergraduate students in the REU program, guiding them through independent research projects focused on market system data collection and competition relationship extraction.
- Supervised undergraduate students from both UT Austin and Northwestern in year-long or semester-long research endeavors on network-based market system engineering and design, assisting in research proposal development, experimental design, and data analysis.

- Coordinated regular meetings to track progress and provide constructive feedback, contributing to successful project outcomes.

- Facilitated collaborative opportunities for undergraduate mentees in the preparation and presentation of conference papers for publication, with one mentee showcasing our work at the 2023 IDETC conference.

Freshman Mentor Jun. 2012 - Aug. 2013

- Program: Freshman Mentorship Program at the University of Electronic Science and Technology of China

- Appointed as a freshman mentor, ranking in the **top 3%** for overall quality, to guide approximately 30 Mechanical Engineering freshmen in their transition to university life and academic studies.

- The major responsibilities include: organizing orientation events, coordinating regular learning activities like seminars and panel discussions, and offering academic guidance to students requiring assistance, etc.

Skills:

- **Languages**: Python, R, MATLAB
- **Frameworks**: Scikit, TensorFlow, Keras, Seaborn, NetworkX, StellarGraph, ergm, igraph
- **Tools**: Gephi, ArcGIS

Service And Professional Membership:

- Reviewer for 21st Annual Conference on Systems Engineering Research (CSER 2024)
- Volunteer of 9th International Engineering Systems Symposium: CESUN 2023

- Assistant reviewer for academic papers and research project reports
- ASME Student Member
- Volunteered as a lab tour guide volunteer for students visiting the Walker Department of Mechanical Engineering at the University of Texas at Austin.