2021

INQUIRIES IN INTELLIGENT INFORMATION SYSTEMS: NEW TRAJECTORIES AND PARADIGMS

Samaa Elnagar
Virginia commonwealth university

Graduate School

INQUIRIES IN INTELLIGENT INFORMATION SYSTEMS: NEW TRAJECTORIES AND PARADIGMS

INQUIRIES IN INTELLIGENT INFORMATION SYSTEMS: NEW TRAJECTORIES AND PARADIGMS

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Business at Virginia Commonwealth University

By

Samaa Elnagar

Master of Business Administration, University of Akron, Ohio, USA, 2017.

Bachelor of Engineering (Electronics & Communication), Alexandria University, Egypt 2007

Directors: Prof. Kweku-Muata Osei-Bryson (Chair)

Dr. Manoj Abraham Thomas (co-Chair)

Department of Information Systems

Virginia Commonwealth University

October 2021

Table of Contents

Table of Figures

Tables

List of Acronyms

AI	Artificial intelligence
IIS	Intelligent information systems
IS	Information systems
DSR	Design Science Research
DS	Design Science
COC	Cognitive Computing
NN	Neural Networks
DL	Deep Learning
ML	Machine Learning
KM	Knowledge Management
KMS	Knowledge Management Systems
DSS	Decision Support Systems
MCDM	Multi-Criteria Decision-Making
DM	Data Mining
EA	Evolutionary Algorithms
KG	Knowledge Graphs
ES	Expert systems
SVM	Support Vector Machines
NE	Neuromorphic Engineering
IoT	Internet of things
IDSS	Intelligent Decision Support Systems
CR	Critical Realism
SLR	Systematic Literature Review
Mask R-CNN	Mask Region Convolutional Networks
DLN	Deep Learning network

QA system	Question answering system
TF	Tensor flow
TFlite	Tensor flow lite
RP	Research problem
Spike-NN	Spike Neural Networks
AHP	Analytical Hierarchical Process
SQ	Service Quality
ISDT	Directive-Prescriptive Theory
DREPT	Explanatory-Predictive Theory

Acknowledgement

This dissertation was written during one of the most challenging periods in my life. I had two pregnancies and two beautiful babies during my PhD journey. I stayed most of the PhD period in Richmond, VA, raising my kids alone. That might seem an impossible task. However, I was lucky to have the support to overcome those challenges. The PhD fruited ten conference papers already published, three journal papers under review, and more in progress.

At first, I would like to thank GOD, the most merciful, supportive, and gracious. God never failed me nor denied my prayers. He finds me a way when there are no possible ways. I had many difficulties during the pandemic, but I would also like to thank my dissertation committee for their academic and personal support and understanding my situation. Dr. Kweku-Muata Osei-Bryson has always encouraged me to learn new technologies and apply complex methodologies. As a result, I learned Python and Deep learning. Also, Dr. Manoj is an extraordinary supportive character who would give me all the material and courses I need to accomplish my work. I can’t also deny how he was inspiring me with new research ideas.

To my beloved husband, Waleed Sharfeldin, my consultant, friend, and partner, supported me emotionally, mentally, and financially. He always encouraged me to aim high and go for the best choice regardless of how it will affect him. I strongly believe that a compassionate partner is a great blessing and powerful support. My deepest gratitude to the faculty members and PhD students. A special thanks to my PhD colleague Ramandeep Sandhu, who helped me and supported me so much personally and academically. I can’t also forget Mauli Dalal; she was a good friend and a caring person even to my kids. I would like specially thank Dr. Jayaraman Vijayakumar for his outstanding support and Austen Goldman for his willingness to help in almost everything.

Finally, I have a huge debt to everyone who contributed to this thesis come true, especially my mother, who came twice to help me during the first months after having babies. This dissertation is dedicated to you, and you are all contributors to its success.

Vita

Samaa Elnagar was born and raised in Alexandria, Egypt. Since her childhood, she has been in love with math and analytical subjects. She got her degree in Electronics and Communication Engineering from Alexandria University in 2007. She had been working as a software engineer for seven years, from 2008 till 2014. Her latest position was as a senior .NET software engineer, especially as a backend developer. Then, Samaa got her MBA from the University of Akron, Ohio, the USA, in 2017. Soon, she joined a PhD program in Information Systems at Virginia Commonwealth University. She currently works a visiting assistant professor at Miami University.

Abstract

Rapid Digital transformation drives organizations to continually revitalize their business models so organizations can excel in such aggressive global competition. Intelligent Information Systems (IIS) have enabled organizations to achieve many strategic and market leverages. Despite the increasing intelligence competencies offered by IIS, they are still limited in many cognitive functions. Elevating the cognitive competencies offered by IIS would impact the organizational strategic positions.

With the advent of Deep Learning (DL), IoT, and Edge Computing, IISs has witnessed a leap in their intelligence competencies. DL has been applied to many business areas and many industries such as real estate and manufacturing. Moreover, despite the complexity of DL models, many research dedicated efforts to apply DL to limited computational devices, such as IoTs. Applying deep learning for IoTs will turn everyday devices into intelligent interactive assistants.

IISs suffer from many challenges that affect their service quality, process quality, and information quality. These challenges affected, in turn, user acceptance in terms of satisfaction, use, and trust. Moreover, Information Systems (IS) has conducted very little research on IIS development and the foreseeable contribution for the new paradigms to address IIS challenges. Therefore, this research aims to investigate how the employment of new AI paradigms would enhance the overall quality and consequently user acceptance of IIS.

This research employs different AI paradigms to develop two different IIS. The first system uses deep learning, edge computing, and IoT to develop scene-aware ridesharing mentoring. The first developed system enhances the efficiency, privacy, and responsiveness of current ridesharing monitoring solutions. The second system aims to enhance the real estate searching process by formulating the search problem as a Multi-criteria decision. The system also allows users to filter properties based on their degree of damage, where a deep learning network allocates damages in

each real estate image. The system enhances real-estate website service quality by enhancing flexibility, relevancy, and efficiency.

The research contributes to the Information Systems research by developing two Design Science artifacts. Both artifacts are adding to the IS knowledge base in terms of integrating different components, measurements, and techniques coherently and logically to effectively address important issues in IIS. The research also adds to the IS environment by addressing important business requirements that current methodologies and paradigms are not fulfilled. The research also highlights that most IIS overlook important design guidelines due to the lack of relevant evaluation metrics for different business problems.

Chapter 1 Introduction

The trend toward data-driven decision-making has created a paradigm shift in how organizations create and leverage knowledge for decision-making (Davenport & Kudyba, 2016). NASA (2011) describes intelligent systems as autonomous, robust, and collaborative systems. According to Gartner's 2018 technology trend survey (S. Li, Da Xu, & Zhao, 2018), Artificial Intelligence (AI) is listed as the No. 1 strategic technology in developing information systems. An Intelligent Information System (IIS) can be considered to be an Information System (IS) that exhibits intelligent behavior that is based on AI (Aronson, Liang, & Turban, 2005; Elhoseny, Metawa, Darwish, & Hassanien, 2017; Ghoshal & Moran, 1996). On the same line, Intelligent Information Systems (IIS) are intelligent systems that can imitate, automate several intelligent behaviors of human beings (Elhoseny et al., 2017) (J. E. Aronson et al., 2005). So, what makes a system intelligent is its ability to infer and perceive beyond the situations it was trained for. Intelligent systems are designed to maintain sustainable performance in changing environments, decrease waste in resource utilization, and guard against catastrophic failure. However, not all systems based on AI are considered intelligent information systems. Otherwise, there are basic characteristics that define IISs, which are discussed further below.

Expert systems, intelligent agents, and knowledge-based systems are examples of IISs that notably enhance productivity and quality (Zhaohao, Ping, & Dong, 2012). IISs involve various types of Information Systems (IS) and technologies, such as Decision Support System (DSS) (Sperandio, Gomes, Borges, Brito, & Almada-Lobo, 2013), Rule-based Expert System (ES), Knowledge Management System (KMS), Supervised Machine Learning (Marshall & Lambert, 2018), Recommender Systems, Deep Learning (LeCun, Bengio, & Hinton, 2015a), Cognitive Computing (Soman & Suri, 2016a), and Internet of Things (IoT).

IIS has been going through foundational longitudinal changes due to the change in the AI algorithms used. In the traditional rule-based computing paradigms, such as supervised machine learning (ML) and clustering algorithms, intelligent systems fall short of self-learn and adjust their behavior to react to changes in the outside environment (Kinsner, 2009). Through the emergence of Deep Learning (DL), AI has entered a new era of cognitive AI or "*new generation AI*" (Duan, Edwards, & Dwivedi, 2019). DL networks offer inference and reasoning features through unsupervised learning" (Duan et al., 2019). DL allow machines to be augmented with human-like cognitive abilities such as NLP, object recognition, and computer vision (S Elnagar & Thomas). The new "*new generation AI*" is the current paradigm for IISs (Ransbotham, Kiron, Gerbert, & Reeves, 2017).

Intelligent systems based on the DL paradigm are goal-oriented systems, not task-oriented systems, but DL stills resource-consuming, and intelligence gained by such networks couldn't be explained as a "black box" (Shwartz-Ziv & Tishby, 2017). While AI (e.g., *Deep Learning (DL)*, *Internet of Things (IoT)*) is the current bedrock of IIS, Cognitive Computing (*CoC*) is the foundation for next-generation IIS, as shown in Figure 1.1. COC is offering transparent AI where intelligence is encoded using special types of mathematics (Emmert-Streib, Yli-Harja, & Dehmer, 2020). In addition, COC's new hardware, *Neuromorphic hardware*, allows low-cost computation and communication, which uses spikes as a method of communication between layers (Aghnout & Karimi, 2019).

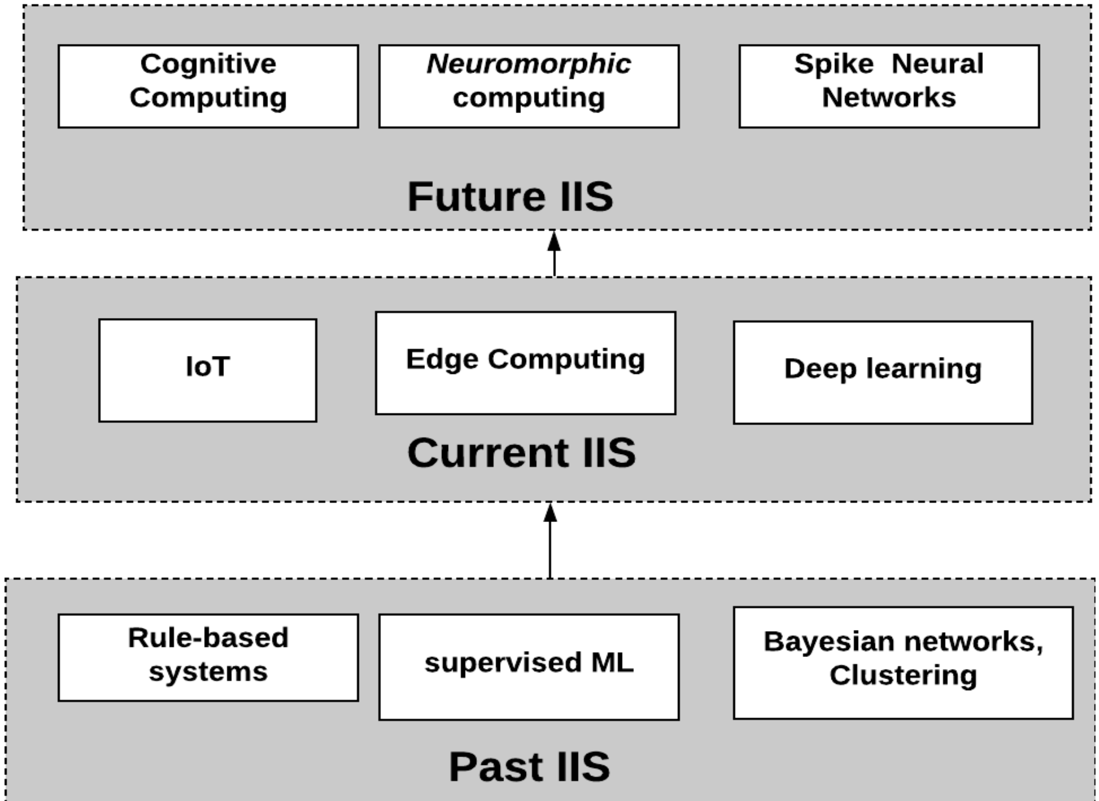

Figure 1.1 Technologies Used in The Longitudinal Development of IIS.

1.1 Characteristics & Challenges of IIS

Krishnakumar (2003) defines intelligent systems as systems that can be characterized by flexibility, adaptability, memory, learning, temporal dynamics, reasoning, and the ability to manage uncertain and imprecise information. Accumulating the characteristics of IIS from research by Crowder, Carbone, and Friess (2020) and (Guerlain, Brown, & Mastrangelo, 2000), we could summarize the high-level characteristics of successful IIS as:

A. *Interactivity*: Intelligent systems are part of the digital ecosystems. The digital ecosystem is an "open, loosely coupled, domain clustered, demand-driven, and self-organizing environment, where each system is pro-active and responsive " (Boley & Chang, 2007). An intelligent system should be able to interact with other systems and with humans.

B. *Event Detectio*n: An IIS should be able to identify and communicate important changes and updates, changes, or any issue that triggers a change in the system (Guerlain et al., 2000).

C. *Communication Skills*: the system communicates information in an informative way. This might involve a specific communication language, such as an information agent's language.

D. *Predictive Capabilities*: The system can predict and infer the effect of actions on future performance, which means to predict both the future environmental state in addition to the change in states caused by different decisions.

E. *Adaptation*: intelligent systems should adapt the change to the environments and change inside themselves too. Within change, intelligent systems are adapting, changing, and aging.

By reviewing the recent trend in Intelligent systems research, we can agree with Pupkov (2019) and Akerkar (2012) that research in IIS faces numerous challenges. This research will address specifically these challenges.

- **Responsiveness**: Interactive intelligent systems often suffer from a basic conflict between their computationally intensive nature and being responsive to the user (Gerring, Shortliffe, & van Melle, 1982). Traditional IISs have suffered from poor system responsiveness, including response time, latency, and false-negative rates (Hizam & Ahmed, 2020). *Responsiveness* has several measures other than *Latency,* such as the awareness to respond to user requests (Jiang, Chan, Tan, & Chua, 2010). Latency is directly correlated with responsiveness. The delayed is the response, the less responsiveness in the system (Ho & Lee, 2007).

- **Privacy**: Currently, intelligent systems have more accessibility to user information more than ever before. Thus, IISs become a potential threat to privacy which raises significant ethical and legal concerns (Michael, Fusco, & Michael, 2008).

- **Interactivity Levels:** Although IISs are inherently interactive, IIS interaction levels are still underdeveloped (Eiband, Buschek, & Hussmann, 2021). There is a prodigious need to develop and upgrade intelligent interactive technologies for use in IISs (Balaž & Predavec, 2018). According to Human-Computer-Interaction research, designing an IIS should consider three important issues: user mindsets, user involvement, and knowledge outcome (Gretzel, 2011). *Flexibility* always coincides with *Interactivity* (Yadav & Varadarajan, 2005). So, supporting website *flexibility* will increase interactivity and consequently better service quality (Pur, 2017) (Jiang et al., 2010).

- **Efficiency:** efficiency is assessed as the resources consumed to accomplish the task, the cost of the e-service, and the quality of the offered service (Hizam & Ahmed, 2020). Most IISs are resource-exhaustive in terms of computation and resources consumption. Maintaining efficient computation is a key challenge, especially in limited devices such as IoTs (L. Zhao, Lu, Zhang, & Chau, 2012).

- **Tangibility:** is the clear reflectiveness of the resources necessary for providing services to customers (Pakurár, Haddad, Nagy, Popp, & Oláh, 2019) (Sun, Teh, & Chiu, 2012). The "*Tangibles*" are the visible aspects of the service quality to improve customer satisfaction (Panda & Das, 2014). *Tangibility* is a primary *service quality* dimension where the service representation gives a clear, concrete image of the service (Santos, 2002).

1.2 Motivation for The Research

The intrinsic motivation of the research is addressing the challenges faced by IIS and their impact on service quality and user satisfaction. The current paradigms offer foreseeable contributions to solve current IIS challenges. Although DL empowers IIS with noticeable cognitive competencies, DL is resource exhaustive and cannot solve uncertainty nor resolve conflicts (Akerkar, 2012). On the other hand, there has been a strong bias towards building intelligent systems over evaluating them. This is mainly due to a lack of universal measures such as efficiency, complexity, and interactivity levels (Gretzel, 2011). Instead, intelligent systems are not evaluated at all or evaluated using subjective qualitative measures. The focus of IISs evaluation is usually on the general aspects of information system success, involving measures such as intention to use or actual use and user satisfaction (DeLone & McLean, 2003).

The lack of proper evaluation metrics has impacted the design of IIS that overlook important metrics to be considered, and eventually, those overlooked metrics will turn to be the challenges faced by IISs. For example, a design that does not consider enhancing the response rate will be suffering a responsiveness issue. The fact that evaluation metrics, especially service quality, differ according to the problem and goal to be achieved (Margaria & Steffen, 2006).

Information Systems Design Science Research (DSR) addressed emerging AI paradigms such as IoT, Edge Computing, and intelligent agents (Chatterjee & Armentano, 2015) (Essays, 2018). These areas are of growing research interest, especially in enhancing the decision-making and Knowledge management processes (Zhaohao, 2019). However, IS, specially DSR lacks research in applying DL to IoTs and edge computing specifically. In this research, two different research problems are addressed; each has its motivation and evaluation metrics. In section 1.3, the detailed motivation of each problem is provided along with its significance, and the objective is provided in the next sections.

1.3 Research Objectives

This research aims to overcome several challenges of IIS that impact service quality and user satisfaction (Ali et al., 2021). The research integrates the cutting-edge paradigms of DL, IoTs, and Edge computing to IIS from a Design Science (DS) perspective. The new paradigms overcome limitations of the rule-based logic that proved difficult to adapt to changes in the environments (Akerkar, 2012). The research proposes two DS artifacts: the first is a multi-criteria decision-making for real estate based on the degree of damage. The second is a scene-aware ridesharing monitoring system using optimized deep learning for the Internet of things. However, each of the proposed research problems is discussed further in detail.

1.3.1 Overview on The Two IIS Research Problems (RPs)

In this subsection, overviews of the two RPs are presented. For each RP, a later chapter will provide more details, including research objectives and also a description of how the problem was addressed, associated results, and performed evaluations. Below we list the titles of the two RPs

1. SAFEMYRIDES: A Scene-Aware Ridesharing Monitoring System Using Deep Learning for The Internet of Things

2. A Multi-Criteria Decision Making for Real Estate Based on Degree of Damage: Towards Enhanced Service Quality.

In presenting the overview of each RP, we will briefly describe its associated motivation, objective, and significance. More details are presented later in the relevant chapters.

1.3.2 SAFEMYRIDES: A Scene-Aware Ridesharing Monitoring System Using Deep Learning for The Internet of Things

Motivation: Embedding Deep Learning Networks (DLNs) to Internet of Things (IoT) applications has been a topic or interest (Leminen, Rajahonka, Wendelin, & Westerlund, 2020), Therefore, many optimization techniques were developed to simplify applying DL to IoTs. Ridesharing services are successful shared economy applications that contribute to a significant share of the transportation economy. However, ridesharing still suffers from serious safety and security issues such as harassment and assaults. Current real-time monitoring systems are costly and complex requiring continuous connectivity to other servers. In addition, continuous on-the-cloud monitoring is prone to security risks and violates user's privacy. Advances in deep learning for the Internet of Things (IoTs) enable regular smartphones (IoTs) to run deep learning models and provide real-time decision support without network dependency.

Objective: This research presents a scene-aware system for monitoring ridesharing vehicles. The system uses optimized deep learning models that run locally on smartphones to detect violations in ridesharing and record violation incidences. The system would enhance customer trust and safety in ridesharing without violating privacy. The system decreases the cost of cellular internet connections and video processing.

Significance: The proposed ridesharing monitoring system would add to the IS design science knowledge by connecting two emergent yet important paradigms: *Edge Computing* and *DL* for *IoT*. Moreover, the research provides insights to many industries that aspire to apply DLNs to sensitive business applications (Shi, Cao, Zhang, Li, & Xu, 2016). The system is efficient in terms of memory, storage usage, and minimal connection to cellular internet networks. Therefore, the system would help to prevent hundreds of crimes and violations, especially in developing countries where computational resources are limited, and high crime rates tend to be higher.

1.3.3 RP2: A Multi-Criteria Decision Making for Real Estate Based on Degree of Damage: Towards Enhanced Service Quality.

Motivation: Searching for real estate properties is a challenging multi-dimensional problem that involves diverse econometric, spatial, temporal, and structural dimensions. Currently, electronic agents such as Zillow and Realtor offer limited spatial filtering for real estate properties, giving little consideration to semantic filters such as the property's degree of damage or State of Maintenance (STM). With the advent of Deep Learning for computer vision, the degree of damage could be detected automatically in real estate images. This research includes the degree of damage as a novel feature for filtering real estate. The degree of damage is detected automatically from real estate images using the deep learning network of Mask-RCNN. Images show the condition of the interior and exterior and indicate damage in different sections of a house (Bin, Gardiner, Li, & Liu, 2019).

Objective: The research aims to develop a real estate decision support that formulates the real estate search process as a multi-criteria-decision problem using Analytical Hierarchal Process (AHP). In addition, the system incorporates the degree of damage as a novel feature for filtering real estate. The degree of damage is extracted automatically from real estate images using the deep learning network of Mask-RCNN (Z. Huang, Zhong, Sun, & Huo, 2019). While damages in real estate images might be hard to allocate, Mask R-CNN can capture the finely detailed objects precisely.

Significance: The system not only enhances the flexibility of the real estate search process but also enhances the efficiency and relevancy of the search process and results. Including the degree of damage in filtering real estate enhances the service quality *tangibility,* end-user experience and, therefore, user satisfaction (Bhardwaj, Di, Hamid, Piramuthu, & Sundaresan, 2018).

1.4 Significance of The Research

This research has a significant impact on society and IS research. Each of the two research adds to IS body of knowledge as each problem is a DS artifact that integrates different cutting-edge technologies in a logically coherent way to addresses some areas of IIS challenges. More importantly, the research adds to the IS environment by realizing important business needs that are not fulfilled by current methodologies and paradigms. The research considers the challenges of IISs as design objectives to overcome current system limitations. In tables 1.1, 1.2, and 1.3 below, we link our two research problems with IIS characteristics, challenges, and research questions.

Table 1.1 Research Problems & IIS Characteristics

Research Problem (RP)	**IIS Characteristics**				
	Int	**EvD**	**CSk**	**ADP**	**PrC**
SAFEMYRIDES: A Scene-Aware Ridesharing Monitoring System Using Optimized Deep Learning for The Internet of Things	√	√	√	√	√
A Multi-Criteria Decision Making for Real Estate Based on Degree of Damage: Towards Enhanced Service Quality.	√			√	√

Int: Interactivity; *EvD:* Event Detection; *CSk:* Communication Skills

ADP: Adaptation; *PrC:* Predictive Capabilities

Table 1.2 Research Problems & IIS Areas

Research Problem (RP)	IIS Areas				
	EC	**DL**	**IoT**	**DSS**	**KM**
SAFEMYRIDES: A Scene-Aware Ridesharing Monitoring System Using Deep Learning for The Internet of Things	√	√	√	√	√
A Multi-Criteria Decision Making for Real Estate Based on Degree of Damage: Towards Enhanced Service Quality.		√		√	√

EC: Edge Computing; *DL*: Deep Learning; *IoT*: Internet of Things

KM: Knowledge Management; *DSS*: Decision Support Systems

Table 1.3 Research Problems & IIS Challenges

Research Problem (RP)	IIS Challenges				
	Rsp	**Prv**	**IL**	**Eff**	**Tang**
SAFEMYRIDES: A Scene-Aware Ridesharing Monitoring System Using Optimized Deep Learning for The Internet of Things	√	√	√	√	
A Multi-Criteria Decision Making for Real Estate Based on Degree of Damage: Towards Enhanced Service Quality.	√		√	√	√

Rsp: Responsiveness; *Prv*: Privacy; *IL*: Interactivity Level

Eff: Efficiency; *Tang*: Tangibility.

1.5 Research Questions

Based on the research motivation, objective issues were discussed earlier. We could formulate the research problem as five important questions. These questions guide the direction and focus of the research. Since we are developing two systems, the research questions are an aggregation of research questions of both systems. The link between the research questions and two research problems is discussed in table 1.4. The research questions are listed below:

RQ1: How could the introduction of decentralizing control at the IoT level overcome the delay, overhead, and cost associated with ridesharing monitoring?

RQ2: Does the proposed ridesharing monitoring system outperform the existing solution in terms of responsiveness and privacy preservation?

RQ3: How could the ridesharing monitoring system improve the quality of ridesharing monitoring service?

RQ4: How could the use of Multi-Criteria Decision Making enhance the relevancy, efficiency, and flexibility of the real estate search process?

RQ5: How could introducing the degree of damage feature enhance the user's satisfaction through enhancing the service quality tangibility and search results relevancy?

These five research questions define the scope of the dissertation. The exploration of these research questions will be addressed within the context of two related IIS research problems (RPs).

Table 1.4 Research Problems & The Modified RQs

Research Problem (RP)	**RQ1**	**RQ2**	**RQ3**	**RQ4**	**RQ5**
SAFEMYRIDES: A Scene-Aware Ridesharing Monitoring System Using Optimized Deep Learning for The Internet of Things	√	√	√		
A Multi-Criteria Decision Making for Real Estate Based on Degree of Damage: Towards Enhanced Service Quality.				√	√

1.6 Conclusion

In summary, this research aims to investigate how the intellectual competencies of current cutting-edge paradigms overcome IISs challenges in terms of enhancing service quality and user satisfaction. Past IIS were considered task-oriented systems that apply rule-based AI and regular neural networks. Currently, *DL, IoT, Edge Computing (EC)* offer a leap in the cognitive competencies of IIS towards goal-oriented systems. This research offers two DS artifacts as an application for DL to IIS. The first is A Scene-Aware Ridesharing Monitoring System Using

Optimized Deep Learning for The Internet of Things. The second is A Multi-Criteria Decision Making for Real Estate Based on the Degree of Damage.

In the next chapters, chapter two provides a literature review is provided for previous research on IIS in addition to an overview of AI, DL, IoT, EC, and Service Quality. Chapter 3 will be addressing the research methodology used in the two research problems. Chapters 4 will give an overview of techniques used to apply DL for IoTs and edge devices in terms of issues and tradeoffs. Chapter 5 will provide the proposed SAFEMYRIDES system and the associated results. Chapter 6 will deliberate the use of Mask CNN to detect damage in real estate photos in terms of network structure and detection algorithms. Chapter 7 will present the proposed A Multi-Criteria Decision Making for Real Estate Based on Degree of Damage and the associated results. Chapter 8 discusses the conclusion of the dissertation.

Chapter 2 Literature Review

In the literature review section, we give an overview of Intelligent Information Systems (IIS). The overview dives into the foundational components of IIS, such as intelligence, cognition, and Artificial Intelligence (AI), to be able to elaborate on how new paradigms will change the behavior and structure of IIS. Next, we will provide definitions, characteristics, of IoT and Intelligent agents.

For intelligence, we will elaborate on the difference between traditional symbolic logic. Next, we provide an overview of AI and develop two different taxonomies of AI algorithms. The first taxonomy is based on the longitudinal development of AI, and the second one is based on the foundation logic of AI algorithms. We are going to focus specifically on Deep Learning (DL) as the most salient AI algorithm to date. Importantly, an overview of Edge Computing, the Internet of things, is presented. Lastly, an overview of Service Quality is given as the venue to evaluate IIS. To understand how different topics are intersected, we depicted the relation between different topics as shown in Figure 2.1.

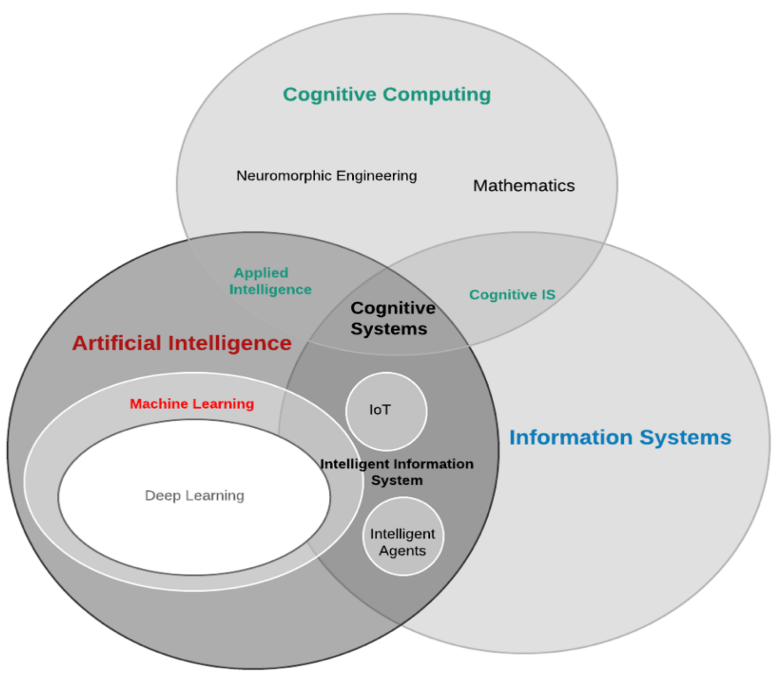

Figure 2.1 Intersection of AI, IIS and Cognitive Computing

2.1 Intelligence

According to the dictionary, intelligence is the capacity for learning, reasoning, understanding, and similar forms of mental activity (Mohamadnejad, Gholami, & Ataei, 2012). So, intelligence is a set of "primary mental abilities". The primary mental abilities are seven according to Thurstone (1938), which are verbal comprehension, verbal fluency, the numerical computation and arithmetical reasoning, spatial visualization, inductive reasoning, memory, and perceptual speed. Cognition means a range of mental processes relating to the acquisition, storage, manipulation, and retrieval of information (Y. Wang et al., 2012) (Gathercole, 1999). Those primary mental abilities can be subdivided into two further kinds, "fluid" and "crystallized." Fluid abilities are the reasoning and problem-solving abilities. Fluid abilities could be measured using tests such as analogies, classifications, and series completions (Cattell, 1971).

However, current symbolic logic intelligence was found to be limited in cognitive functions. In addition, traditional IIS is hard to adapt and evolve. To learn how cognition is implemented in the brain, we must build computational models that can perform cognitive tasks and test such models with brain and behavioral experiments.

2.2 Artificial Intelligence

According to the dictionary, Artificial Intelligence (AI) is the capacity of a computer to perform a range of operations analogous to learning and decision making in humans. AI is strategically transforming many IT systems (Daugherty & Wilson, 2018). IIS exhibits its intelligence by achieving goals in the face of different changing environments (Simon, 1980). Currently, there are limitations in the current AI technologies (conventional AI technologies) specially in dealing with dynamic environments because they are providing limited inference, while it is expected that next-generation AI could evolve and adapt to change the environment autonomously (Samaa Elnagar & Weistroffer, 2019). Conventional AI systems can't explain the reasoning process of decision making, nor how to solve the Blackbox issue, i.e., knowing why decisions are made in a certain way (Castelvecchi, 2016).

AI has gained a lot of taxonomies, such as a *longitudinal evolution taxonomy* (Dartnall, 2013), based on the timeline of the advancement in AI algorithms, and a *taxonomy based on the algorithmic paradigm* (Golstein, 2018) as discussed below.

2.2.1 AI Longitudinal Evolution Taxonomy

There is no agreeable longitudinal evolution taxonomy of AI. However, based on Duan et al. (Duan et al., 2019) and (Golstein, 2018), the evolution of AI has always been interrupted by the emergence of breakthrough AI technologies. AI has gone through about four phases of development. Till 1980, AI was only based on systematic rule-based methods. The second phase development of AI was from 1980 to 2000. By the emergence of Machine Learning (ML) based methods (supervised, unsupervised, reinforced learning) and other analogy methods, AI could perform more sophisticated classification decisions. Some of the salient AI algorithms for this phase are Support Vector Machines (SVM) and Neural Networks (NN) (Chang & Lin, 2011) .

The third phase from 2000 to 2020 was highlighted by the development of the breakthrough Deep learning algorithms (unsupervised ML algorithms that have inference abilities) (LeCun et al., 2015a). AI technologies such as speech recognition and image recognition highlight this development phase. The fourth phase is still under research which is *cognitive AI*. The fourth phase will emerge under *cognitive computing and Neuromorphic Engineering* development (Soman & Suri, 2016). The longitudinal evolution taxonomy of AI is presented in Table 2.1.

Table 2.1 Longitudinal Evolution Taxonomy of AI

Phase	**Disruptive Technology**	**Highlights**
Till 1980	Systematic rule-based method	Expert systems (ES). Accounting systems
From 1980 to 2000	Machine Learning (ML), Bayesian networks, Clustering	Support Vector Machines (SVM) and Neural Networks (NN) (Chang & Lin, 2011)
from 2000 to 2020	Deep learning	Speech recognition and image recognition (LeCun et al., 2015a)
Future	cognitive computing, Neuromorphic Engineering	Cognitive Agents (Soman & Suri, 2016a)

2.2.2 AI taxonomy based on Algorithmic paradigm.

AI algorithms differ widely in their logic and practice. However, algorithms that follow the same logic have been clustered to what are called "AI tribes (Domingos, 2015). AI algorithms are mapped into five tribes based on paradigm classification. *Symbolism* is a Logic-based approach where logical reasoning or inverse deduction is used to drive conclusions. *Symbolism* has been widely used in traditional AI IIS, such as Knowledge management systems (KMS) and Expert systems (ES). *Connectionism* or machine learning such as neural networks and deep learning is inspired by the mammalian brain. Machine learning is an AI model where the machine is learning through training, and the machine adjusts its weights through many iterations while propagating back the error to adjust to the new weights. *Connectionism* has gained a lot of attention recently because of the inference capabilities offered by deep learning.

Evolutionism performs search and optimization such as Genetic algorithms and swarm intelligence that have been used extensively in many IIS in the last decade. The *Bayesians* are probabilistic methods that use probabilistic inference to justify prior hypotheses. Interest in *Bayesian* networks has been renewed as they are used in autonomous vehicles. Finally, the *Analogy-based* methods that perform *Clustering* of similar groups. *Analogy* algorithms such as SVM and K-Nearest Neighbor have been powerful AI algorithms used in many IIS such as KMS, DSS, and ontologies. *Analogies* are used basically with Knowledge-based methods since they try to extrapolate from existing knowledge and previous similar cases. The AI taxonomy based on the algorithmic paradigm is presented in table 2.2.

Table 2.2 AI Taxonomy Based on the Algorithmic Paradigm.

Algorithmic paradigm.	**Rationality**	**Algorithms**	**IS application**	**Ref.**
Symbolism **(Inductive Logic Programming)**	Using inverse deduction from knowledge composition	Rule-based	KM-DSS- DM, Expert systems	(Bowonder & Miyake, 1992; Rekik, Kallel, Casillas, & Alimi, 2018)
Bayesian	Using probabilistic inference to justify the prior hypothesis	Bayesian networks	autonomous vehicles	(Y. Zhao, Tang, Darlington, Austin, & Culley, 2008)
Connectionism	weight assignments and backpropagation of errors	NN	KM-DSS	(Yaqoob et al., 2016) (Frajberg, Bernaschina, Marone, & Fraternali, 2019)
		DL	NLP (Sentiment analysis, KD)	
Analogy **(**Clustering)	Using kernel machines (supervised or unsupervised) to group similar samples)	K-Nearest Neighbor (KNN)	Data analytics, DM, KM	(Ragini, Anand, & Bhaskar, 2018; Yaqoob et al., 2016; Zantout & Marir, 1999)
		Support Vector Machines (SVM) kernel machines.	Recommender systems, phishing detection, ontologies	
Evolutionary Algorithms (EA)	Structuring (mimicking living organisms' behavior)	Genetic algorithms	DSS, Database systems	(Kennedy, 2006; Lebib, Drias, & Mellah, 2017)
		Swarm intelligence	DSS, Recommender systems	

2.3 Deep Learning

Deep learning has various definitions; one of them is that "Deep learning is a set of algorithms in machine learning that attempt to learn in multiple levels, corresponding to different levels of abstraction. It typically uses artificial neural networks, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts." (Wikipedia, 2019). Conventional neural network techniques cannot process natural data in their *raw* form and require careful engineering to design a feature extractor that transforms the raw data into a proper representation or feature vector. Only using a feature vector the learning subsystem, often a classifier could detect or classify patterns in the input (Schmidhuber, 2015)

Deep learning discovers the intricate structure in large data sets to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer (Deng & Yu, 2014). A deep-learning architecture is a multilayer stack of simple modules and many of which compute *non-linear* input-output mappings. Each module in the stack transforms its input to increase both the selectivity and the invariance of the representation (LeCun, Bengio, & Hinton, 2015a). Deep learning is making major advances in increasing the *inference* abilities of machines. Currently, DL can solve problems that have resisted the best attempts of the artificial intelligence community for many years.

From the above overview, we can conclude that Deep learning converts raw data without supervision into tacit knowledge. The tacit cognitive knowledge is represented in the weights and connections between input, hidden, and output layers. The technical tacit refers to the tacit knowledge used for inferences and classification, also known as the *Process knowledge (Know-How)*. Although DL requires less domain expertise, explaining and codifying the knowledge in the learned models is difficult. Therefore, DL knowledge cannot be directly transferred. Otherwise, we have to move the DLN itself to the device. However, many industries and business sectors are aspiring to apply DLNs to edge and end devices for the state-of-the-art inference capabilities that DLN offer (K. Li, Daniels, Liu, Herrero-Vinas, & Georgiou, 2019). A review of the most popular DL networks is provided.

Examples of DL models include convolutional neural networks (CNN) and recurrent neural networks (CNN). *Deep Neural Networks (DNNs)* are *Feed-Forward Networks (FFNNs)* or one-way networks between the input layer to the output layer without going backward. *Recurrent Neural networks (RNNs)* include connections between passes and connections through time. *RNNS* could be an *FFNN* with a time twist. Connections between nodes in *RNNs* form a directed graph as features links from a layer to the previous layers, allowing information to flow back into the previous parts of the network, allowing more information persistence (Xiaobin Zhang, Chen, & Huang, 2018).

Convolutional Neural Network (CNN) is a class of deep neural networks that mimics the human eye cortex in recognizing images and is most commonly applied to analyzing visual imagery. *CNN,* unfortunately, cannot detect multiple objects at a time. So, a *Regional Convolutional Neural Network (R-CNN)* can force the CNN to focus on a single region at a time. Before using CNN for classification and bounding box regression, the regions in the R-CNN are resized into equal size by a selective search algorithm. Therefore, it helps to specify a preferred object (J.-C. Huang, Huang, & Liu, 2019).

2.4 Internet of Things

According to Leminen et al. (2020) and O'Connor, Rowan, Lynch, and Heavin (2017), the Internet of Things (IoT) refers to 'smart' devices, often with sensor capability, that are able to collect, share, and transfer data using the Internet (p. 80). In simple words, IoT is to make everyday objects "smart" by equipping these "things" with sensors, processors, and wireless communication capabilities. Smart thermostats and wireless door cameras are a part of the Internet of Things (IoT) ecosystem. IoT is not only a smart device but also a hugely valuable amount of big data.

Managers are increasingly excited to adopt the Internet of Things (IoT) which allows big data generated from users to be included in the decision-making processes. The use of IoT might yield many benefits for organizations specially in the manufacturing and healthcare fields (Boos, Guenter, Grote, & Kinder, 2013). IoT is important for asset management in organizations because they provide enough quality data to help asset managers make the right decisions at the right time

(Brous & Janssen, 2015). For example, IoT can be used to detect traffic jams, and IoT traffic sensors could help find multiple forms of transportation using location sensors.

The remote connection capability of IoT made it easy for a manager to monitor and control the performance of sensitive machines from a distance through means of ambient intelligence (Ramos et al., 2008). IoT adoption also affects the social and phycological aspects of the organizational structures (Brous & Janssen, 2015), where many tasks performed by people become automated, while new tasks have emerged, such as data science engineer.

Adoption of IoT in DSS design allows more detailed and accurate predictive analysis, increasing trust in the asset management process and allowing greater predictability in risk-based decisions IoT partially increases automation in decision-making due to the greater certainty in IoT measurement as to when and which action needs to be taken (Lunardi, 2016).

2.5 Service Quality

Intelligent systems are part of the digital ecosystems. The Technology Acceptance Model (TAM) and IS success model have been used extensively in IIS-related studies. There is a great bias toward investigating the intentions to use and no enough research on the functional use (Venkatesh & Davis, 2000). The challenges of IISs discussed earlier could be incorporated under the umbrella of enhancing service quality. Service Quality (SQ) differs widely based on the type of service, type of provider, and the type of outcome. There are many service quality models for websites, mobile applications, and IoTs (Hizam & Ahmed, 2020). Studies found that service quality has three dimensions (interaction quality, environmental quality, and outcome quality). These dimensions have significant and positive effects on the accumulative satisfaction (Ali et al., 2021), while only one dimension of service quality (interaction quality) has a significant and positive effect on transaction-specific satisfaction (L. Zhao et al., 2012).

(Margaria & Steffen, 2006) proposed a model that posits service quality as a sequence of components. The quality of the physical service environment (during a service encounter) and the process-based quality: which refers to the quality of the physical service encounter. The overall quality is the sum of the quality of the these sequences. Over the years, many researchers have identified and developed measurement scales to assess the websites (Parasuraman, Zeithaml, &

Malhotra, 2005). WebQual was built on 12 dimensions consisting of *information fit to the task, interaction, trust, response time, design, intuitiveness, visual appeal, innovativeness, flow, integrated communication, business process, and substitutability* (Loiacono, Watson, & Goodhue, 2002). WebQual TM has concentrated on the operationalization quality and the technical quality of the websites rather than the service quality provided to customers through the websites.

One of the most important SQ dimensions is the "tangibles" dimension of services, which has been insufficiently explored and defined. The "tangibles" dimension refers to the firm's use of technology that affect the impression gained by the Internet user. Does the site have pictures of each item shown in detail? Is sufficient information provided on each item? (Sun et al., 2012)**.** For these reasons, the "tangibles" dimension is hypothesized to be robust in explaining the overall service quality specially for Internet searchers (Seiler & Reisenwitz, 2010)**.**

For IoT, (Hizam & Ahmed, 2020) proposed the IoT-SERVQUAL model of service quality for the Internet of Things (IoT) that measures service quality for the Internet of things based on four dimensions (i.e., *Privacy, Functionality, Efficiency, and Tangibility*). (E. Y. Huang, Lin, & Fan, 2015) developed M-S-QUAL, a Mobile service quality framework with five factors (*contact*, *responsiveness*, *fulfillment*, *privacy,* and *efficiency*). Jun and Palacios (2016) found that the accuracy, ease of use, and continuous improvement, are considered the main sources of customer satisfaction/dissatisfaction in using mobile applications.

Since the dimensions of service quality are variable based on the service type, enhancing one or more of the dimensions will consequently enhance the overall customer satisfaction and trust, as shown in Figure 2.2. Selecting only relevant dimensions to a research problem is the key to enhance the overall SQ and customer satisfaction. The definition of each of the dimensions used in the research will be given later in the implementation chapters.

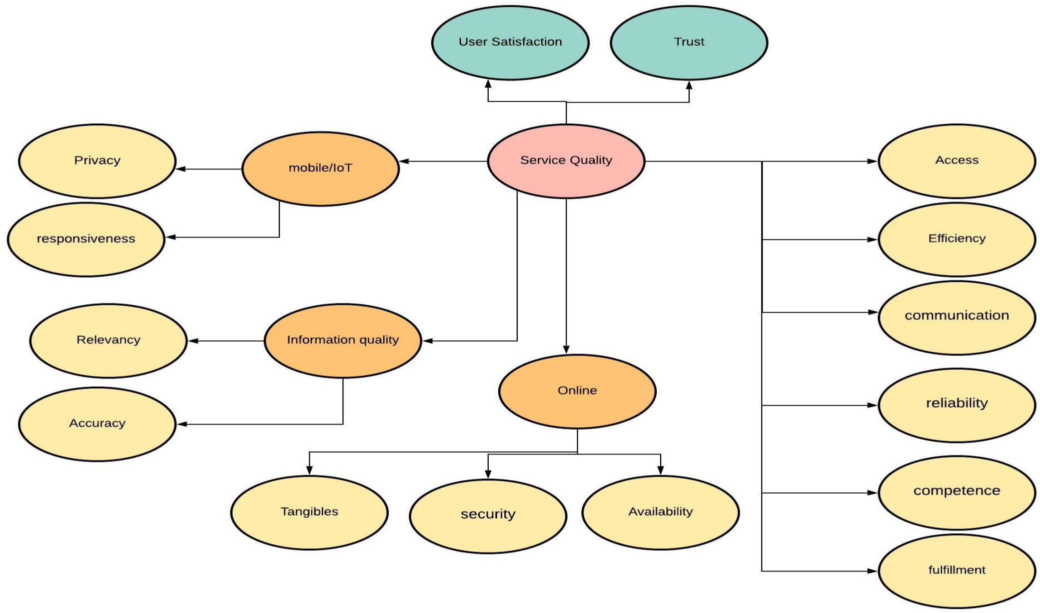

Figure 2.2 Service Quality Dimensions Including Online Websites, Mobile and IoT.

2.6 Summary

We can conclude from the previous literature review that since IIS is the intersection between IS and AI, as in Figure 2.1, the style of intelligence used in developing IISs defines their performances. Traditional symbolic logic cannot overcome uncertainty, and it is found to be limited in providing a wide range of intelligent functions. While DL provides strong inferences and reasoning capabilities, it is still a Blackbox method, and an exhaustive resource method. So, DL needs to be optimized before they are used in everyday IoTs applications. Service quality is a key measure to evaluate and design IIS. SQ differs widely based on the type of service, type of provider, and the type of customer. So, it is preferable to choose relevant dimensions to a given problem.

Chapter 3 Research Methodology

The research methodology is the most important aspect that shapes and guides research studies (Rubin & Babbie, 2012). Iacono, Brown, and Holtham (2009) defined research methodology as the efficient use of techniques and procedures in the research design. Research methodology is used to establish a grounded approach that structures the research study, such as strategy, approach, research philosophy, and methodology components.

The research methodology can be classified in different ways. The most common and widely used methods are the qualitative and quantitative approaches (Kothari, 2004). Moreover, selecting the research methodology is usually a multifaceted multidimensional process that uses mixed instruments. However, selecting the appropriate IS research methodology is not an easy task because researchers have to consider the aims of the research, theories, evaluation, and technologies (Al Kilani & Kobziev, 2016). In the following sections, we will present different classifications of research methodologies.

3.1 Classification of Research Methodology Based on Objective

The most popular classification of research methodologies is classifying research based on the objective: Design Science Research (DSR) or Behavioral Science (BS) (A. Hevner, S. T. March, J. Park, & S. Ram, 2004). According to Hevner (Hevner & Chatterjee, 2010), BS seeks to develop and justify theories that explain or predict the analysis, design, implementation, and use of information systems. These theories are affected by design decisions, while DS aims to build artifacts, such as methods, constructs, models, and instantiations. In this research, we are building two DS artifacts.

3.2 Classification of Research Methodology Based on Analysis

This classification is widely adopted across most sciences: qualitative research versus quantitative research. Qualitative research methods often rely on informative or critical social sciences to help researchers study social and cultural phenomena (Crabtree & Miller, 1992). The qualitative approach provides explanations to explore a particular phenomenon and build theories through data collection and analysis, shifting the philosophical assumptions to appropriate research design and data techniques. Qualitative research has been used widely in IS using ethnography, action research, case research, interpretive studies, and examination of documents and texts (Palvia, Mao, Salam, & Soliman, 2003). The quantitative research uses statistical analysis methods to answer questions such as "how many?", "How often?" or "to what extent?" (Newman, Benz, & Ridenour, 1998).

3.3 Classification of Research Methodology Based on Philosophy

There are many popular research philosophies such as *Positivism, Interpretivism, Empiricism, Realism, Relativism* (Demetis & Lee, 2016). Most behavioral science research in IS following *Interpretivism or Positivism* (Kuhn, 1970). However, each philosophy has its limitations that lead to adopting new philosophies such as Critical Realism (Carlsson, 2006).

3.4 Approaching Research Design

The central focus of this research is to build two IIS that apply current AI paradigms and investigate how these paradigms can overcome some challenges and enhance service quality. The two IIS are design science artifacts, but they differ in orientation, technologies, and output. The first research aims to develop a scene-aware ridesharing monitoring system. The second research aims to enhance the real estate search process and introduce a new degree of damage filter. Thus, the design science research methodology and outputs used in the two research are different, and we have a distinct methodological design in conducting each research.

3.5 Selected Research Methodology

As discussed earlier, the research methodology has many orientations based on the nature of the problem addressed. The two systems are considered design science artifacts according to IS DSR. The DS paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts (A. R. Hevner, S. T. March, J. Park, & S. Ram, 2004). However, the design methods used in each problem differ widely in terms of the type of DSR artifact produced. So, both problems share similar DSR principles but differ in the output type, philosophy, instruments, and evaluation. In the following section, we will be discussing every aspect in detail.

3.6 Design Science Research Methodology

The science of design, as theorized by Simon (1969), is a study of the "systematic creation of knowledge about and with design (pg. 441). The determinative objective of DSR is the design of novel artifacts. The DSR artifact contribution in this role corresponds to Simon's (1996) inner (or interior) environment of the artifact. Essential research activity in a DSR project is the reflection, learning, and design theorizing around the development of artifacts. The project captures prescriptive design knowledge from the DSR design cycles and the introduction of the artifact into the problem space of the project. The developed knowledge can be represented in multiple formats as knowledge for technology, action, integration, or process.

3.6.1 Scope of Design

The scope of the design could be either on a local or an abstract level. The design of a local-level artifact created for a specific context will be through a combination of action and design research (Sein, Henfridsson, Purao, Rossi, & Lindgren, 2011). Then, researchers address its generalizability beyond the singular local context. The design of abstract-level artifacts is defined for a well-defined class of contexts. Then, researchers guide how to instantiate and customize the abstract artifact to specific contexts (Gregor & Jones, 2007). For designing abstract artifacts, researchers' informing challenges are more numerous. The researchers need to convincingly demonstrate the artifact's potential utility (its conceptual relevance) (Kinsner, 2009). However, the artifacts need to be customized to every organization to align with the organization's strategy and culture.

We consider the decomposition of the artifacts to small connected artifacts as the pathway towards a potentially creative design process. Decomposition is driven by a collaboration between design analysis and research through design. The first research is a ridesharing monitoring system, and the scope of the design is abstract. That's means that the system is not designed for a certain ridesharing agent. Rather the system is designed to monitor a basic function in any ridesharing company. The system is decomposed into two sub-artifacts. The first is the full-size trained DL network that can detect violations in ridesharing. The second is the smartphone application that applies the DL network to monitor ridesharing sessions.

The second system aims to enhance the search process for any real estate website. The system is built upon the basic search tools found in most real estate agents. So, the research design is also abstract that aims to be applied to most real estate websites. The system is decomposed into three sub-artifacts. The first is a trained DL network that can detect damage in real estate images. The second is the MCDM algorithm, and the third is the real estate search system that combines the two other artifacts.

3.6.2 Relevance

Nicolai and Seidl (2010) characterize three types of relevance in management research outcomes: *Conceptual, Instrumental, Legitimate. Instrumental relevance* simply verifies if the technical instruments used in the research are relevant to solve the research problem. *Conceptual relevance* is the most popular type of relevance. The artifact can only realize its actual utility after the respective audiences decide in favor of its adoption for a specific context, based on its perceived conceptual relevance. A. R. Hevner et al. (2004) associated relevance with the perceived utility "The relevance of any design-science research effort concerns a constituent community" pg 85. Scientific knowledge can also serve as a means of credentialing specific knowledge domains. However, the actual utility is part of the evaluation of the proposed artifact. Users of DS artifacts will decide the relevance of the artifacts in enhancing the creation of innovative ideas.

During the design phase, we choose *Instrumental Relevance.* In the ridesharing monitoring system, the drivers' smartphones are the main devices in ridesharing sessions. The designed system utilizes the cameras of the driver's smartphone. Moreover, the trained deep learning model to detect violation are compressed and optimized to run on the limited computational power of smartphones. In the second research, the choice of MASK-RCNN was relevant to the nature of damage detection where damage needs to be specially allocated and evaluated. Moreover, since the real-estate search involves many criteria of preference, choosing AHP for MCDM is a relevant choice because AHP is simple and intuitive and lets the user specify criteria of preference easily.

3.6.3 Theories

(Hart & Gregor, 2010) distinguished five interrelated types of theory: (1) theory for analyzing, (2) theory for explaining, (3) theory for predicting, (4) theory for explaining and predicting, and (5) theory for design and action. Another popular taxonomy of theory developed by Iivari (1990) who defined three levels of theories in IS: (1) at the conceptual level, at which the objects of inquiry are defined (2) a descriptive level, at which the explanatory conjectures and hypotheses are generated and tested; and (3) a prescriptive level, at which methods for constructing systems are put forward, with recommendations for their practical use.

However, the role of theory in design science has gained a lot of debate. We agree on the importance of theory in designing DS artifacts. Still, we disagree with limiting the design to a confined theoretical perspective, trying to fit the design to a certain theory, or trying to fit the theory into the design. Otherwise, a theory is not a rigid formation but a malleable architecture that changes according to design settings and outputs. Studies have found that kernel theories are at such a high level of abstraction that their relationship to design is difficult to discern. Therefore, we are following the view of (Kuechler & Vaishnavi, 2012), where theory is not only a design input but a design output along with the artifact itself.

A design science artifact produces mid-range theories that both describe and prescribe the kernel theory in the context of the design science artifact. Mid-range theories are defined as "Theories that lie between the minor but necessary working hypotheses that evolve in abundance during day-to-day research and the all-inclusive systematic efforts to develop a unified theory" (Merton & Merton, 1968). According to Kuechler and Vaishnavi (2012), The mid-range theories are divided into *directive-prescriptive theory (ISDT) and explanatory-predictive theory (DREPT).* An *ISDT* is a set of prescriptive statements describing how a class of artifacts should behave (meta-requirements).

Kernel theories are used in the design product and design process to suggest either the meta requirements or the construction process (Arnott, 2006), while DREPT captures a different sort of design-related knowledge. DREPT is considered the translation of highly abstract constructs from natural, social, or design sciences fields to the realm of artifact-achievable effects. DREPT could be summarized as:

1. A type II-IV (explanatory/predictive) theory, according to Gregor (2006) typology.
2. They were derived from an abstract kernel theory that was initiated in a non-design science domain.
3. In DREPT, the kernel or tacit theory has been translated into the technology domain.

In other words, DREPT maps kernel theory (technology-free theory) to a design-related mid-range theory that can explain why a designed technological artifact. So, the design of the cognitive ideation framework accommodates both a priori theorizing and a reflective after-development theory. We believe that kernel theories are useful for guiding and explaining the framework's design and giving it a more solid grounding in design science. Still, theories are not fully developed until the framework is built and evaluated. That aligns with Kuechler and Vaishnavi (2012) assumption that every fragment of knowledge is situation-based and cannot be predicted from theoretical considerations in advance. The knowledge is a part of a working design to clarify the implications of the theory in a given circumstance.

In the design of the ridesharing monitoring system and the real estate MCDM system, we build the conceptual frameworks as a summary of the gap in the literature. The conceptual frameworks act as initial mid-range theories that shed light on the guidelines for designing the artifact.

3.6.4 Evaluation

Aken (2004) suggests that the evaluation activities could be designated as 'experimental production' and 'experimental control' (Bhaskar, 1998). IS evaluation approaches have traditionally paid limited attention to the designed and developed system after its deployment or initial implementation. Both research problems will be evaluated based on how they could enhance service quality (SQ), as discussed in section 2.6. However, the SQ dimensions evaluated in each research problem are selected based on the relevancy of the dimensions to the research problem itself. The implication of this method is to build multiple theories from each artifact. However, the first research is evaluated using instantiations and controlled experiments. The second research is evaluated using instantiations and illustrative examples. A detailed plan of the evaluation process is discussed in chapters four and five.

3.6.5 Design as a Search Process

According to (A. R. Hevner et al., 2004), the DSR process entails exploiting different methods to reach the desired output while fulfilling theories in the problem environment. Similarly, the design of IIS artifacts is accomplished by searching for the best methods that can solve the problems. Both of the designed systems selected the technical methods based on how they can overcome the limitations of methods used in the literature review.

3.6.6 Rigor

Rigor is achieved by appropriately applying existing foundations and methodologies. In the ridesharing monitoring system, we used the methodology of optimizing DL networks in order to minimize the size and weights of the DL network to fit the computational limitations of IoTs. The algorithms used for DL compression and quantization are discussed further in chapter four. In the real estate MCDM system, we used the methodology of preference elicitation to help users define their criteria of preference, including the novel criterion "degree of damage." Further details are in chapter five. The summary of the DSR research activities of the two proposed systems are guidelines is summarized in table 3.1:

Table 3.1 DSR Guidelines-Based Activities of This Research.

	Ridesharing Monitoring System	**Real Estate MCDM System**
Description	Develop a monitoring system for ridesharing	Develop a real estate search system
Role to IIS	Applying new IIS paradigms	Applying new IIS paradigms
DSR output	Instantiation	Framework
Design theory	Research Problem-based hermeneutics	Research Problem-based hermeneutics
Design instruments	DL networks. Edge computing, IoT, Android	DL networks, MCDM, web development
Evaluation	Prototype, Instantiation	Instantiation, Illustrative Examples
Research Rigor	The framework is based on the clear gap in the literature that calls for privacy preservation and secure rideshare session	The framework is based on a clear gap in the literature that calls for an enhanced real estate search process in terms of flexibility, relevancy, and efficiency
Research Relevance	Using Instrumental relevance through choosing technical instruments relevant to solve the research problem	Using Instrumental relevance through choosing technical instruments relevant to solve the research problem
Design as a Search Process	Searching for the best instruments or the best DL optimization method for trained DL that detected violations	Searching for the best instruments or MCDM algorithm to solve the research problem
Research Contributions	the system would help prevent hundreds of crimes and violations, especially in developing countries where computational resources are limited, and high crime rates tend to be higher.	The system enhances the interactivity of the real estate filtering process and enhances the relevancy of the search results. Including the degree of damage in filtering real estate enhances end-user experience and, therefore, user satisfaction.

Chapter 4 SAFEMYRIDES: A Scene-Aware Ridesharing Monitoring System Using Optimized Deep Learning for The Internet of Things

4.1 Introduction

Ridesharing services have gained increasing popularity in the last decade, becoming the first choice of private transportation in many countries (Levinger, Hazon, & Azaria, 2020). Safety is still a significant concern to address in ridesharing (Feeney, 2015). There have been reports of harassment, assault of passengers and drivers on these rides (Eisenmeier, 2019; Garcia & O'Brien, 2019). However, no strict measures are taken due to challenges imposed on the driver, vehicle, and road infrastructures (Chaudhry, El-Amine, & Shakshuki, 2018).

While ridesharing has reduced alcohol-related fatalities and diverts some drinking groups away from using a designated driver strategy (Miller et al., 2020; Weber, 2019), people under the influence are prone to many harassment and theft crimes (Dills & Mulholland, 2018). Some countries in Europe and the developing parts of the world report ridesharing to be comparatively unsafe (Chaudhry et al., 2018). This suggests the dire need of having safety measures to protect both the driver and the passenger (Hong, 2017).

The broad research problem is addressing the safety concerns in ridesharing through monitoring ridesharing sessions. Satisfaction of ridesharing users is driven by service quality, perceived usefulness, trust, and safety (Arteaga-Sánchez, Belda-Ruiz, Ros-Galvez, & Rosa-Garcia, 2020; Aw, Basha, Ng, & Sambasivan, 2019). According to Ulrich Beck's concept of risk society (Beck, Lash, & Wynne, 1992), surveillance risk are mitigated by safety avails offered. Therefore, using monitoring systems in ridesharing would enhance safety and consequently increase user satisfaction. However, there are many challenges to implementing ridesharing monitoring.

From the social and governance aspects, recording the entire rideshare session violates the passenger's privacy (Anderson, 2016; Z. Lee, Chan, Balaji, & Chong, 2016). Technical challenges include storing and processing ridesharing monitoring sessions, in addition to overcoming latency in detecting violations. Moreover, cellular communication in developing countries, where crime rates are higher, is limited (Dillahunt, Kameswaran, Li, & Rosenblat, 2017). On the one hand, one research direction aims at enhancing the automatic processing of surveillance videos on the cloud by enhancing the network connection using 5G technologies, vehicular communications applications, and smart cities infrastructure for media processing (Guevara & Auat Cheein, 2020). However, smart cities' infrastructures adopt a centralized decision-making approach through shared computational resources or edge units, raising many privacy and security concerns (E.-K. Lee, Gerla, Pau, Lee, & Lim, 2016).

Edge computing solutions move the processing of the ridesharing monitoring to nearby edge units. In attempts to solve the latency and cloud dependency issues, edge computing-based monitoring solutions focused on increasing the transmission bandwidth, optimizing workloads, and increasing the number of edge processing nodes (T. Wang, 2020). However, these solutions not only necessitate costly special infrastructure but also impose latency risks. In high traffic, the latency issue would be doubled due to the increased processing workload dedicated to each edge unit (Meshram, Choudhary, & Velaga, 2020; Xingzhou Zhang, Wang, Lu, Liu, & Shi, 2019). Therefore, edge computing-based solutions partially solve the security issues, but the privacy, latency, and network dependency issues are not entirely solved.

Even though studies have emphasized the increasing significance of monitoring ridesharing sessions (Almoqbel & Wohn, 2019), little research has focused on moving the monitoring decisions to end devices (smartphones). Furthermore, minimal research focused on decentralizing decisions through the combination of optimized deep learning (DL) models and IoT (Samaa Elnagar & Thomas, 2020). Decentralizing control and federating the computation process at the IoT level will minimize security and latency risks that emerge from transferring data to other computation nodes. In addition, decentralizing decisions will ensure privacy where ridesharing monitoring will be constrained on a single local device (Y. Li & Taeihagh).

Recently, smartphones' specifications as the Internet of Things (IoT), are continuously enhanced while the cost is fairly decreasing (El Khaddar & Boulmalf, 2017). The enhancements in the computational resource of smartphones inspired applying deep learning for end devices (Samaa Elnagar & Osei-Bryson, 2020). Optimized deep learning models could run on the limited processing units, batteries, and memory to provide stand-alone decision-making and recognition tasks (Samaa Elnagar & Thomas, 2020).

This research aims to bridge the gap between ensuring safety, privacy, and security in ridesharing monitoring without network dependency, costly infrastructure, or sacrificing latency. The research introduces a local scene-aware monitoring system for ridesharing services using smartphones. The system detects only violation incidences instead of recording the entire rideshare session to ensure passenger privacy. The system applies state-of-the-art deep learning optimized models to regular smartphone devices to detect ridesharing violations.

Unlike existing research on ridesharing monitoring, the proposed system doesn't depend on continuous internet connection or cloud/edge processing. Moreover, the proposed system decentralizes control and moves the computation and the decision-making to the end device level to decrease latency and the cost of detecting violations. To ensure security, the system encrypts the detected violations before submitting them to ridesharing agents. Encryption will avoid security risks in transferring such confidential data to the cloud or edge nodes. The implementation of the system exhibits superior predictive performance and allows the identification of violation incidence at run time with the least detection latency among current solutions.

The research aims to solve the safety and privacy risks associated with ridesharing by detecting the violations of the common ridesharing code of conduct. The system is motivated by the lack of real-time monitoring systems in ridesharing, the high cost (connectivity cost, processing cost), and the preach of privacy of the current solutions (continuous monitoring of passengers). The system would enhance customer satisfaction and the quality of the ridesharing applications by ensuring safety, privacy, and security. The system also could be applied in developing countries where high computational resources and mobile internet connection are limited. Based on the goals discussed above, we pose three research questions (RQ) that frame the research:

1. How could the introduction of decentralizing control at the IoT level overcome the delay, overhead, and cost associated with ridesharing monitoring?
2. Does the proposed system outperform the existing solution in terms of responsiveness and privacy preservation?
3. How could the proposed system improve the quality of ridesharing monitoring service?

In the following sections, we review prior research on ridesharing monitoring. Then, an overview of ridesharing violations, followed by the methodology and description of the proposed system. The architecture of the system is described and evaluated experimentally and using illustrative scenarios. Finally, future work and limitations are provided.

4.2 Literature Review

4.2.1 Ridesharing Issues

Monitoring ridesharing sessions has been a controversial issue. Y. S. Lee (2018) conducted a study on *Lateral Surveillance* in Singapore. They found that contrary to the belief that individuals are always skeptical of surveillance, the respondents, on average, reported a positive perception of both top-down and lateral surveillance. They also found that an individual's appraisal of a surveillance system is directly related to the amount of risk the surveillance system mitigates for them.

Z. Lee et al. (2016) propose a benefit-cost framework that mainly focuses on the privacy and security risks to explain the deterrents for active participation in the sharing economy. However, earlier studies pay little attention to broader concerns such as the safety of passengers, which cause consumers to not participate in sharing economy services (Hong, 2017). (Cheng, Bao, Zarifis, & Mou, 2019) developed a model of consumer trust in ridesharing platforms from the perspective of psychological contract violation (PCV) and subsequent service recovery efforts of these platforms. The study investigates how the type and magnitude of the violation moderate the relationship between psychological contract violation (PCV) and consumer trust in the ridesharing platforms.

4.2.2 Previous Ridesharing Monitoring Systems

Previous ridesharing monitoring systems include the Safe-Share Ride system developed by L. Liu, Zhang, Qiao, and Shi (2018). The system can detect violations happening in the vehicle by detecting a predefined list of seeking help phrases. The system uses speech recognition for verbal help detection. Then, the system starts recording videos and analyze them on the nearest computational points. However, depending on the seeking help phrases for violation detection is not typical in most cases, especially for passengers under influence. (Gupta, Buriro, & Crispo, 2019) introduced Driver-Auth, a multi-modal biometric-based authentication for drivers' identity.

The system utilizes three biometric modalities, i.e., swipe, text-independent voice, and face, to verify the identity of registered drivers. However, the research didn't address the detection of violations.

Most current research is dedicated to improving the latency and accuracy of video processing on the cloud or edge computational units. Video Analytics in Public Safety (VAPS) is one of the most successful edge computing applications (X. Zhang et al., 2019). But due to the high real-time requirements and communication overhead, it is hard to implement VAPS at scale or in developing countries. (Ma et al., 2017) designed an efficient caching strategy based on the measurement insights to cash videos to be efficiently transferred and processed.

(Long, Cao, Jiang, & Zhang, 2017) developed a cooperative video processing scheme using an edge computing framework to enable cooperative processing on resource-abundant mobile devices for delay-sensitive multimedia IoT tasks. In the research, several edge nodes cooperate for enhanced video task preprocessing performance. (Yi et al., 2017) presents the LAVEA system, an edge computing platform, which offloads computation between clients and edge nodes, and collaborates with nearby edge nodes. The research aims to provide low-latency video analytics at places closer to the users to minimize the response time.

(L. Wang, Zhang, Li, Zhong, & Shi, 2019) designed MobileEdge, a three-task offloading system that shares computing tasks to nearby on-road computational units. The system target achieving optimal task scheduling for collaborative computing. The system used *Tensorflow lite* that is used in our research. The results show that MobileEdge significantly reduces the application response latency. However, on-road computational units exist only in certain areas in developed countries, Moreover, collaborative computing with other computational units is prone to serious security threats and privacy violations.

(Ran, Chen, Zhu, Liu, & Chen, 2018) developed DeepDecision, a mobile deep learning framework for edge video analytics. The framework ties together computationally weak front-end devices with more powerful back-end helpers to allow deep learning to choose local or remote execution and determine an optimal offload strategy. However, sharing computation imposes security and privacy violations. In addition, the system assumes a well-defined shared computation infrastructure.

(R. Kumar, Mukherjee, & Singh, 2017) used smartphones for monitoring roads through sensing the road surface and conditions from a moving vehicle using fuzzy logic-based road surface roughness classification. The system collects accelerations of users; processes the information and communication them to a central server. Then, a georeferenced database is updated based on collected information and visualization on Google® maps. However, the research addressed road safety rather than the safety of shared rides or the interaction between passengers and drivers. To obtain a broader picture of the current status of ridesharing monitoring, Table 4.1 provides a comparison between previous systems and the proposed system in terms of methodology and service quality issues.

Table 4.1 A Comparison Between Pervious Systems and The Proposed System in Terms of Methodology and Service Quality Issues.

Research	Methodology	Computation level	Latency	Privacy	Safety
(R. Kumar et al., 2017)	Fuzzy logic-based	Central Server	-	-	yes
(Ran et al., 2018)	Mobile deep learning,	Local or remote execution	yes	-	-
(L. Wang, Zhang, Li, Zhong, & Shi, 2019	Collaborative computing, Tensorflow lite	Nearby on-road units	yes	-	-
(Long et al., 2017), (Yi et al., 2017), (X. Zhang et al., 2019)	Edge computing	Nearby edge nodes, or Shared IoT	yes	-	-
L. Liu et al. (2018)	Speech recognition for verbal help detection	At IoT (speech recognition), edge unit (video)	yes	-	yes

Proposed system	Tensorflow lite, Optimized deep learning	At the IoT(smartphone) level	yes	yes	yes

Thus, the gap in literature could be summarized that many studies have confirmed the safety risks and the lack of trust in ridesharing (Cheng et al., 2019; Z. Lee et al., 2016). However, few solutions were offered to solve these issues. Also, current ridesharing monitoring solutions are oriented towards optimizing the video processing on the cloud or nearby edge nodes. Another line of research is dedicated to the processing of surveillance media using edge analytics where multiple edge computational units are collaborating to decrease the latency of media processing. On the other hand, ridesharing monitoring solutions, that share the computation with on-road units, are violating the privacy and security of passengers/ driver data (Cheng, Su, Luo, Benitez, & Cai, 2021). Building on the road computation units also requires costly infrastructure that is hard to be implemented in developing countries and rural areas. Finally, none of the current ridesharing monitoring systems are trying to move computation locally to the IoT level to decrease network reliance and ensure privacy. The proposed system emphasizes the privacy of customers/ drivers and shares information only in case of detected violations.

4.2.3 Violations in Ridesharing

According to most ridesharing companies, there is an agreeable code of conduct to be followed by both passengers and drivers. A common code of conduct includes (Armant & Brown, 2020; BILL): Avoid sitting next to the driver on the passenger seat. Avoiding any physical interaction with the driver. Physical interaction with the driver includes violent actions, seductions, and arguments. Moreover, the driver cannot sit next to the passenger in the back seats (Macmurdo, 2015). A Driver must wear a seatbelt all the time. The code of conduct also illegalizes the use and hold of weapons during ridesharing sessions. In addition, the code of conduct condemns any sexual conduct between the driver and passenger (Lyft, 2019). In the design of the ridesharing monitoring system, we trained the system's deep learning model to detect these types of violations.

4.3 Conceptual Framework

service quality models help to manage the system capabilities and its resources to provide the clear visibility and usability of the services to the consumers. Many service quality models addressed the evaluation of mobile apps and electronic websites. (Sharma & Lijuan, 2015) found that information quality and online service quality were the key determinants for user satisfaction and sustainability of electronic systems. The basic E-S-QUAL scale comprises 22 items in four factors: *efficiency, fulfillment, system availability, and privacy* (Parasuraman et al., 2005). (E. Y. Huang et al., 2015) developed M-S-QUAL, a Mobile service quality framework with five factors (*contact*, *responsiveness*, *fulfillment*, *privacy,* and *efficiency*). Jun and Palacios (2016) found that accuracy, ease of use, and continuous improvement, as the main sources of customer satisfaction or dissatisfaction in using mobile applications.

Hizam and Ahmed (2020) and (M. Singh & Baranwal, 2018) proposed Quality of service (QoS) models for IoT. *Computing, Communication and Things* are three pillars of IoT. So, enhance the QoS of each pillar is a possible way to enhance the QoS of IoT (M. Singh & Baranwal, 2018). QoS of Communication has the most significant threats on IoT service quality. Problems in the network connection are the time, cost of network, availability, and security as shown in figure 4.1 a. These are issues emerge basically from the continuous reliance on network communication. Network communication in is used for transporting real time data to be processed on the cloud or nearby edge units. QoS of Computing aims to enhance the responsiveness, security and privacy of data processing. QoS of things aims to optimize resources consumption and enhance the efficiency and the accuracy of the IoTs. As shown in Figure 4.1, we can find that privacy, responsiveness and security are shared through more than one pillar of the IoT service quality, raising concerns about the significance of continuous communication on the service quality of ridesharing monitoring.

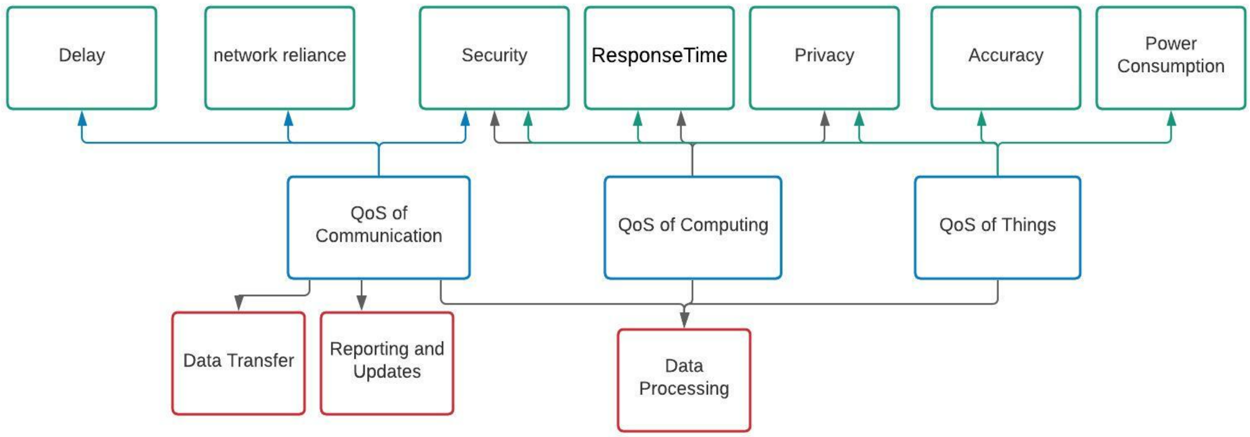

Figure 4.1 a: Quality of Service Issues for Using IoT in Ridesharing Monitoring

Reflecting on the ridesharing monitoring issues discussed in the literature section, we could emphasize that the lack of privacy, safety, and late system response are the issues that affected customer satisfaction and trust.

Most of current research focus on enhancing the issues related to QoS without solving the main root of these issues which is data processing on the cloud or edge units. Our approach is to move data processing to the IoT itself, so network communication is limited to reporting purposes. In other words, the monitoring process is performed on the IoT level and the system only use network connection to report violations to ridesharing agents. Moreover, many researches avoided moving computation to the IoT level because of the limited computational resources of IoT. Therefore, ensuring system *efficiency* in terms of battery, memory, and storage consumption is crucial. This approach will be enhancing the service quality in terms of *privacy, accuracy, efficiency, and responsiveness,* as key factors to enhance ridesharing monitoring service quality and hence customer satisfaction and trust. Each service quality measure is discussed below in detail.

Privacy: gauges the degree to which the application protects customer information. Privacy implies when a business keeps the client data secure and only with their consent, disseminate it to other business activities (Hizam & Ahmed, 2020). Surveillance has been connected with many issues such as Data capitalism. Preach of privacy (West, 2019). Since mobile devices are accessible to sensitive information including biometric features, privacy has been an essential factor in assessing electronic applications (Wolfinbarger & Gilly, 2003; Zeithaml, Parasuraman, & Malhotra, 2002). In the proposed system, the passenger/driver won't be recorded unless a violation incidence is detected. Moreover, recorded media (photos or videos) will be encrypted before they are saved.

Efficiency: measures the resources and ease with which a system is accessed and used (Zeithaml et al., 2002). The efficiency dimension in the proposed system aims to consume fewer resources by applying DL optimization techniques to the violation detection network. For example: the quantization of DL networks improves the overall efficiency in several ways. It saves the maximum possible memory used in the inference process. The Compression of DL networks is decreasing the size of the network to few megabytes, makes it efficient to be used in low storage devices (Jacob et al., 2018). Moreover, efficiency will be evaluated by the resources consumed such as memory, storage, and power consumed.

Responsiveness: is one of the most important service quality dimensions for human-computer-interaction (HCI) (Ho & Lee, 2007). Responsiveness is how the system delivers the results of an operation to users in a timely and organized manner. Latency is directly correlated with responsiveness, the delayed is the response, the less responsiveness in the system. Therefore, the proposed system will be evaluating the average latency to detect a violation.

Accuracy: there are different definitions of accuracy. In the context of this research, accuracy is defined as the ability of the system to detect violations correctly out of all violation incidences. The confusion matrix of precision and recall is also a principal part of calculating accuracy.

4.4 Methodology

In this research, we follow the design science approach (A. R. Hevner et al., 2004). The ridesharing monitoring system is considered a design science artifact of type framework (Peffers, Rothenberger, Tuunanen, & Vaezi, 2012). The artifact is implemented to show functionality. The system is evaluated as instantiation using synthetic or real-world situations to evaluate the detection accuracy, latency, privacy, and consumed resources. So, the evaluation is performed using *controlled experiments* or the study of the artifact in a controlled environment for qualities.

Next, an overview of *Mobile Net Single Shot Detector Network (Mobile SSD)* and different deep learning networks optimization methods is given.

4.4.1 Mobile Net Single Shot Detector Network

In Mobile Single Shot Detector (Mobile SSD), the core layers are built on depth-wise separable filters. The SSD approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes. SSD is designed to be independent of the base network, and so it can run on top of any base network such as VGG, YOLO, MobileNet (W. Liu et al., 2016). The Mobile SSD is composed of two components:

1. The first component extracts feature maps where features are represented in a hierarchal format with incrementing level of granularity from fine to course getting in-depth in the network.
2. the second applies a convolution filter to detect objects, where each object is compared to other objects' feature maps and matched with the closed object.

Mobile SSD takes one single shot to detect multiple objects within the image, while regional proposal network (RPN) based approaches such as the R-CNN series need two shots, one for generating region proposals, and one for detecting the object of each proposal. Thus, SSD is much faster than two-shot RPN-based approaches (Chiu, Tsai, Ruan, Shen, & Lee, 2020).

4.4.2 Deep Learning Networks Optimization

DL models are known for their computational cost and complexity, so they are mostly run on servers on the cloud (J. Wu et al., 2018). On the other hand, *end devices (e.g.,* smart sensors*)* and *edge devices* (e.g., routers) are often battery-powered, have limited memory, processing, and energy resources to store and process data. Applying *DL* models to *IoT* is challenging; DLNs have to be optimized and compressed to fit IoTs limited computational sources. In addition, the optimization of the IoTs themselves is also necessary in terms of memory and hardware optimization (Synced, 2017).

Compression is one of the main optimization techniques that aim to reduce the massive size of *DL* networks. One of the popular compression methods is the *Pruning* technique that eliminates the connections between neurons to directly reduce the feature map width and shrink the network size.

Quantization is another optimization technique that aims at compacting the number of bits required to store the *DLN* weights, usually from 64 bit to 8 bits (Han, Mao, & Dally, 2015).

4.5 System Design

The proposed system is an on-site decision support system for monitoring ridesharing vehicles, where the driver's smartphone can detect the violation and decides the proper action. We trained a Mobile SSD network to detect violation incidences. Then, the trained Mobile SSD is optimized to be deployed to smartphones. The design of the proposed system is shown in Figure 4.1 and the components of the system are discussed below in detail.

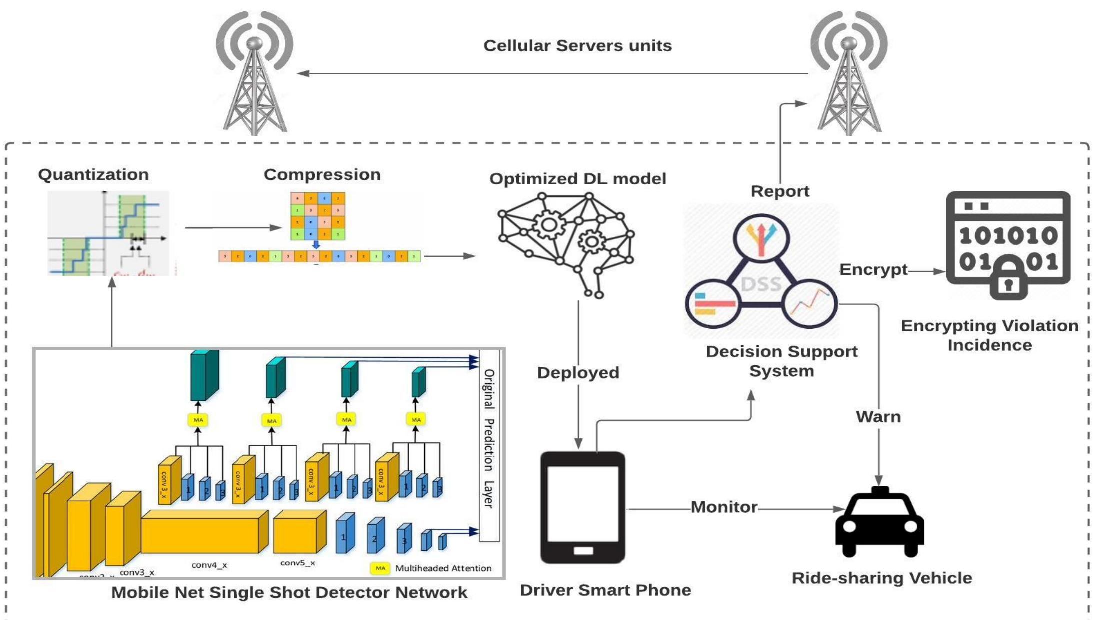

Figure 4.1: Decision Support System for Monitoring Ridesharing Vehicles.

4.5.1 Optimized Deep Learning (DL) Model

Usually, Deep learning models (DLMs) consist of many connected layers resulting in millions of parameters and weight. So, running DLMs tends to be a resource-intensive process in terms of energy consumption and memory accesses and hardware accelerators. To apply a DLM to IoTs and small devices such as smartphones, DLMs must be compressed and quantized. Also, the DLM running on IoTs are read-only means that these compressed models cannot be retrained or edited. The DLM is quantized where parameters are represented in 8 bits instead of 32 bits to accelerate the inference process. The DLM will have a post-training quantization which is the most commonly used form of quantization. In this approach, quantization takes place only after the model has finished training. Moreover, the network is compressed to be 10x less in size of the trained network following the method presented by (P. Singh, Manikandan, Matiyali, & Namboodiri, 2019).

4.5.2 The Ridesharing Smart Phone

Recently smartphones' specifications are enhancing rapidly at a very reasonable cost. The average processing power of most smartphones currently could host an optimized DLM and run inference smoothly. In order for the system to run smoothly, the smartphone should have a minimum RAM (not less than 4GB) and an acceptable resolution camera according to TensorFlow Guide[1].

4.5.3 Decision Support System (DSS)

The DSS is built inside the smartphone. It is responsible for taking action in case of detected violation. If a violation is detected, a warning is voiced to passengers/drivers. If violations continue to be detected, the DSS will record the incidence, encrypt and save the incidence in a hidden folder in the ridesharing. Then, the incidence is reported to the ridesharing agent and attached with encrypted media.

4.5.4 Encrypting Violation Incidence

For security, images or videos that document the violation incidences should be encrypted and saved in the Ridesharing's smartphone storage to ensure passenger/driver privacy. In addition, encryption ensures secure transfer to the ridesharing agent. Otherwise, violation evidence could be deleted before it is reported to the ridesharing agent. On the other hand, encryption will ensure security if the media is shared with nearby road units. So, passengers' information will be kept

---

[1] https://www.tensorflow.org/lite/guide

confidential without driver intervention. Moreover, passenger/driver data is secured in case the phone is stolen or Hijacked.

4.5.5 Transferring Violation Evidence

After the violation incidence is securely recorded and encrypted, it will be uploaded to the nearest cellular server units. In case of no cellular connection, Violation incidences could be uploaded to the nearest on-road units in smart cities infrastructure or sent in text messages.

4.5.6 Cellular Server Units (CSU)

CSUs could be the driver cellular network or the networks that exist on the roadsides in developed countries. So, mostly CSUs are the nearest access point where the DSS could upload incident videos even if no cellular internet is turned off on the phone.

4.5.7 The Ridesharing Vehicle

Information about the ridesharing vehicle is saved in the ridesharing agent database. If the violation requires legal authorities' interventions, the ridesharing vehicle could be found easily. In addition, the ridesharing agent requires registered vehicles to have a valid title, proper condition, and valid insurance for the protection of both the driver and passenger.

4.6 Implementation

The implementation section could be divided into two sections. The first section implements the detection Mobile SSD network where we collect, annotate, and pre-process relevant ridesharing violation samples. Then, the network is optimized (compressed and quantized) to be easily embedded in the DSS. The second section is implementing the DS system that runs locally on the driver's smartphone. The DSS uses different technologies to take action when the optimized Mobile SSD is detecting violations. The DSS actions include sending warnings, encrypting violation incidences, and reporting them to the ridesharing agent. In the following sections, the implementation of each section is discussed in detail.

4.7 Implementing the Trained Mobile SSD Network

4.7.1 The Training Dataset

There is no available dataset that contains images of ridesharing violations incidents. The dataset used to train the Mobile SSD model includes violation images retrieved from web searches, ridesharing agents' articles, and public ridesharing violations videos. The violation images were a collection of different violation types such as violence, physical interaction, and seduction. The images collection included images taken during the day and at night. We also included images for regular ridesharing positions where the interaction between the passenger and driver is with no violation.

4.7.2 Annotation and Preprocessing:

Images were annotated in three classes: driver, passenger, and violation. Violation images included the five-standard code of conduct rules such as interacting physically with the driver as specified earlier. Annotation was performed using Roboflow[2]. Dataset was divided into 90% training, 6 % validation, and 4 % testing. Images were annotated and saved in TFRECORD format. Every class in the image was surrounded by a bounding box representing a driver (pink box), passenger (green box), and violation (purple box), as shown in Figure 4.2. Noted that all passengers are annotated if the image has more than a passenger. Image preprocessing included *resizing* pictures to 416 x 416, *Auto-Orientation,* and *Auto-Adjust of Contrast* through *contrast stretching* (Munteanu & Lazarescu, 1999).

4.7.3 Hardware and Software

The Training was performed on the cloud Google Collaboraty using Jupiter notebooks and to accelerate the training process we used one GPU. The network was trained using TensorFlow 1 and a Keras version less than 2.0 as recommend for custom models training settings[3]. The ridesharing DS system was created using Android studio 4.1.2 and SDK 28. The device used to run building and debugging is a Mac-book pro with an intel core i5 2.5 GHz processor, 8 GB Ram.

---

[2] https://app.roboflow.com/
[3] Model Customization

The build and debug of was performed on a Google PIXEL XL 1 AVD before the system was deployed to a smartphone.

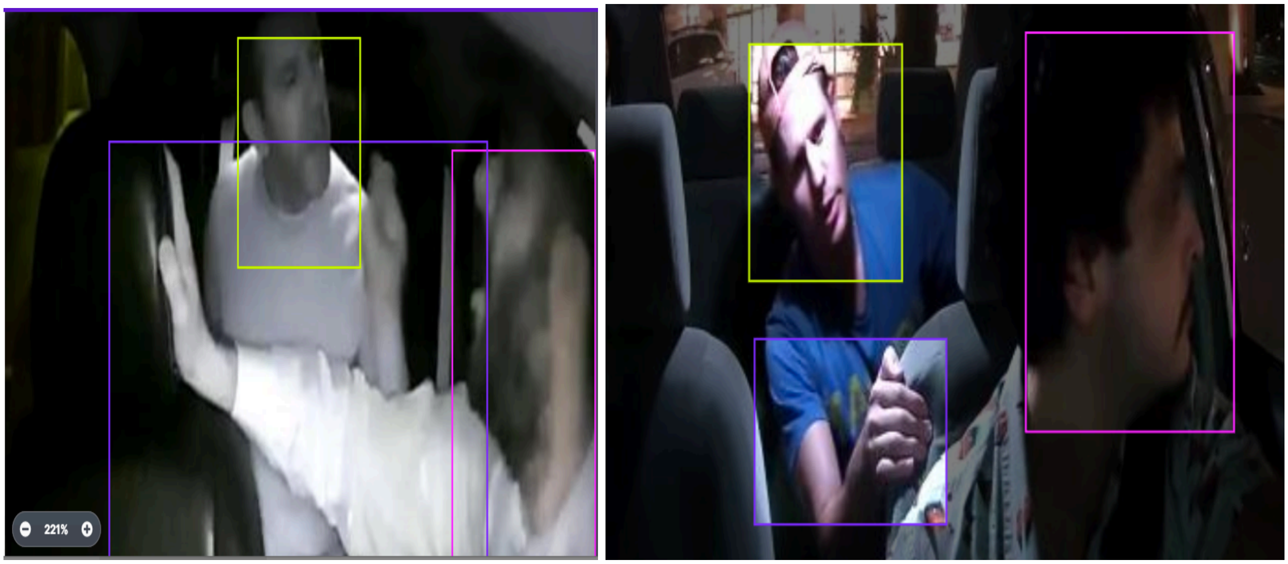

Figure 4.2: Annotated Training, Validation Images Where Each Class is Surrounded with A Bounding Box.

4.7.4 Optimized MobileNet Single Shot Detector Network

We trained a MobileNet Single Shot Detector (v2) (ssd_mobilenet_v2) that is popular for performing object detection tasks on IoT devices. *ssd_mobilenet_v2* is optimized for lightweight inference, enabling it to perform well natively on compute-constrained mobile and embedded devices. The Mobile SSD is quantized to parameters size of 8 bit. Quantization consumes less memory bandwidth. Fetching numbers in the 8-bit format from RAM requires only 25% of the bandwidth of the standard 32-bit format. On the other hand, quantizing neural networks results in 2x to 4x speedup during inference. We didn't train the network from scratch; rather, we transferred learning from a pre-trained model with a COCO dataset. COCO dataset Training weights were uploaded to the training network in the form of .h5 format. Also, we used the same configuration for MSCOCO Dataset as shown below in table 4.2.

Table 4.2: Trained Network Configuration

Parameter	**Value**
The number of layers	6
Minimum network scale	0.2
Max Network scale	0.95
Number of steps	100000
Activation Function	RELU_6

After the network was trained, we ran an inference test, and then we compressed the network and quantized it to 8-bit. Then, the trained network was converted into *TFLITE* format. The 8-bit quantization format is used to run inference on IoTs such as smartphone devices. The average time spent per training step is **global_step/sec:4.96517.** So, the entire training session duration lasted 13.7921 hours.

4.8 Implementing the SAFEMYRIDES Decision Support System

4.8.1 Detection Response

The DSS was developed in Java for Android using Android Studio IDE. The DSS primary interface runs a scene view stream of the smartphone camera. The stream is not recorded. Otherwise, the detection runs on the fly using the smartphone camera running scene. The system has access to both the front and rear cameras. When a violation is detected, the system speaks out "*Violation Detected*" using the *Text to sound* android library. If another violation is detected, the system saves the current scene into an encrypted image file and stores it in a hidden folder in the ridesharing's phone. The average inference time of the system is 28 ms for detecting the three classes of (passenger, driver, and violation).

4.8.2 Encrypting Violation Incidences

For encryption, each violation image is converted into a byte array; then, an XOR operation is applied on each value of the byte array using a predefined encryption key (2-bit key). After performing the XOR operation on each byte array value, the encrypted image is written to a new image file in a jpeg format. However, the driver won't be able to read the encrypted image or decrypt it. Only the ridesharing agent has the predefined key and can decrypt the image. The encryption key acts as a password to encrypt and decrypt the violation image.

4.8.3 The Ridesharing Smart Phone

We ran experiments on two different smartphones with different specifications to monitor the performance of the proposed system on different hardware specifications. The first smartphone used during experiments was a Samsung s10 + phone running on Android. And the second phone is an LG V30+ also running on Android.

The system was built upon the starter code offered by android for object detection using *TensorFlowLite*[4]. The system added the custom trained Mobile SSD network. The code was adjusted to detect the three classes of driver, passenger, and violation. We added a *text-to-speech* module to send violation warnings. In addition, we added the encryption-decryption module for securing violation images.

4.8.4 Results

These are some screenshots from the ridesharing violation application running on a Samsung s10 + phone. The application can detect three classes. A detecting bounding box surrounds each detected class. The number on the right top of the bounding box is the network confidence score in making this prediction. Examples of detected violations by the *SAFEMYRIDES* are shown in Figure 4.3. a, b. The confidence score of the DL network measures how confident is the network assigning an object a specific class.

4.9 Evaluation

The evaluation will be performed on different levels, as shown in table 4.3. The evaluation of the trained Mobile SSD network will be functional. The network is evaluated before and after it is

---

[4] GitHub Code

compressed and quantized. Functional evaluation metrics include testing and training accuracy. We also traced the network testing and validation loss. The system itself is performed through controlled experiments where evaluation relativeness is absolute (e.g., test if the artifact achieves its goal?). Then, another evaluation is performed by conducting a comparison of the artifact performance on a different device of different hardware/software specifications (Cleven, Gubler, & Hüner, 2009). The stages of evaluation performed are summarized in table 4.3.

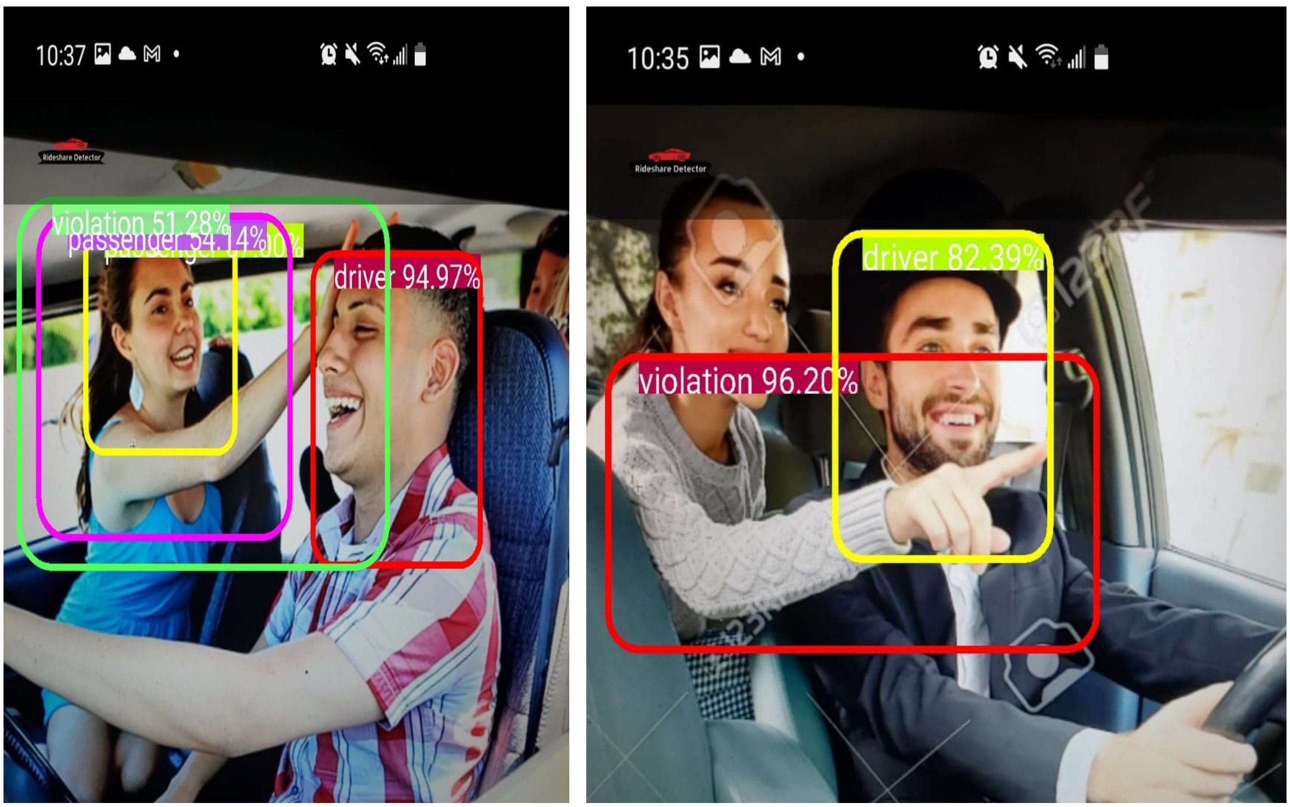

Figure 4.3: a,b a Violation of Sitting Next to The Driver, Interacting Physically with the Driver.

Table 4.3: Evaluation Stages with Involved Components and Criteria.

Artifact	Evaluation Method	Assessed Criteria	Level of Evaluation	Form of Evaluation	Relativeness of Evaluation
Trained Mobile SSD	Reporting the functional performance of the trained network	Training accuracy Loss/validation accuracy Loss,	Instantiation	Quantitative / measured	Absolute
Proposed System	Demonstration of the performance of the artifact with several real examples	Goal, Accuracy, Latency	Instantiation / Real examples	Quantitative / measured	Absolute
Proposed System	Observing the differences between performance on different devices	Comparison between accuracy, latency efficiency on different devices	Instantiation	Quantitative / measured	Relative

4.10 Evaluating The Trained Mobile SSD Network

4.10.1 Training Loss/ Validation Loss

There are two critical measures to evaluate the Mobile SSD: the training loss and the validation loss. The loss is the cost function used while training that needs to be minimized. At the end of every 100 training steps, the network is evaluated for the loss in training and validation. As long as the loss decreases, that means that the network is trained correctly, and it can generalize well (Jancsary, Nowozin, & Rother, 2012). It is expected for the validation loss to be slightly less than the training loss. The network is overfitting If the gap between training loss and validation loss increases. Across the 100K steps of training iterations, the training loss vs. validation loss is plotted as shown in Figure 4.4. The training and validation loss start high and start decreasing gradually. The gap between the training and validation loss is tight. However, at some steps, the validation loss is less than the training loss, which indicates that the validation samples are easier to detect than training samples. The iteration with the least loss is selected in step 96800 with a validation loss of 1.0199343. Then validation loss starts increasing afterward, which means that the network tends to overfit.

The MAP (Mean Average Precision) is another metric used in measuring the accuracy of object detecting networks. The average precision computes the average precision value for the recall values over 0 to 1 using IoU (Intersection over the union) (Henderson & Ferrari, 2016). the MAP was calculated at the end of each 10000 training steps at IoU=0.5 the maximum MAP achieved was **0.38166666 at step 55867. The DetectionBoxes_Precision/mAP = 0.38166666, and DetectionBoxes_Recall/AR@10 = 0.43333334.**

4.10.2 Training and Validation Accuracy

We ran an inference test on the trained network before we performed optimization: compression and quantization to study the effect of the optimization on accuracy and latency. The full-sized mobile SSD model sized 245 MB with a 32-bit inference parameter. The highest training accuracy achieved is 0.94546, and the highest validation accuracy is 0.93123. The average time spent by Google Collaboratory for inference testing is 3.56 seconds.

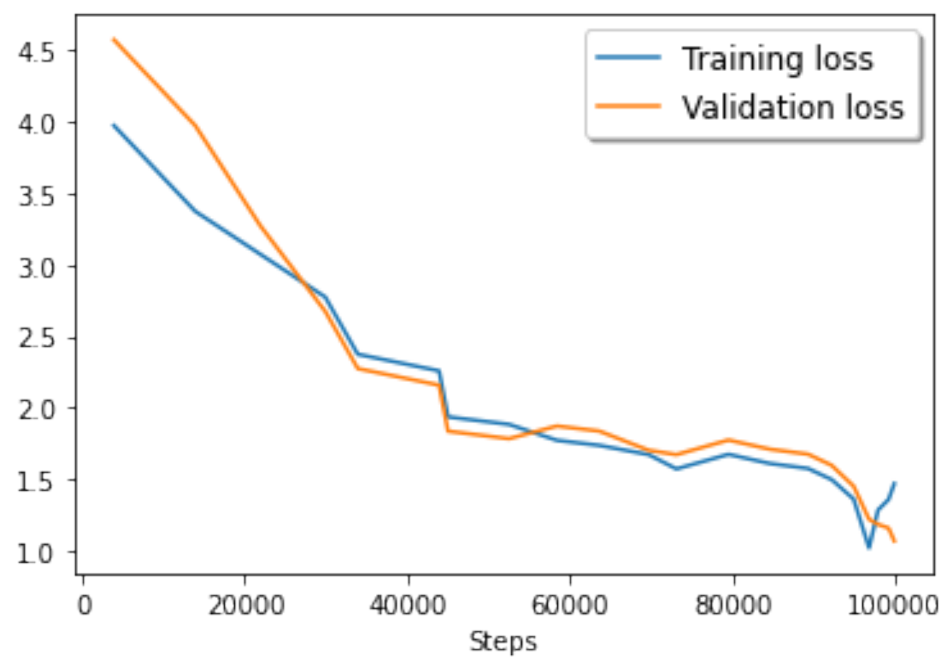

Figure 4.4: Training Loss Vs. Validation Loss of the Trained Mobile SSD.

4.11 Evaluating the SAFEMYRIDES Decision Support System

Firstly, the entire SAFEMYRIDES system was evaluated using a controlled simulated environment where the system was applied in a real ridesharing setting. The first experiments were running on the Samsung S10+ phone. The driver/ passenger performed different violation scenarios such as: interacting physically with the driver, violence, and sitting next to the driver. Examples of detected violations are shown in Figure 4.5 a, b.

4.11.1 Latency

The average system *Latency* is also measuring the system responsiveness that affects service quality and customer satisfaction. Latency would be represented by the average system response time to detect a violation. However, the response time is affected by the surrounding conditions. For example: the response time to detect a violation in daylight is different than the response time at night. After setting the minimum required confidence level to 0.80 (the confidence score of the Mobile SSD that the detected incidence is a violation), the average response time to violations detection during daylight is 450 ms while at night is 790 ms. However, this average response time is higher than other systems used in ridesharing processing, as shown in table 4.4 and Figure 4.6.

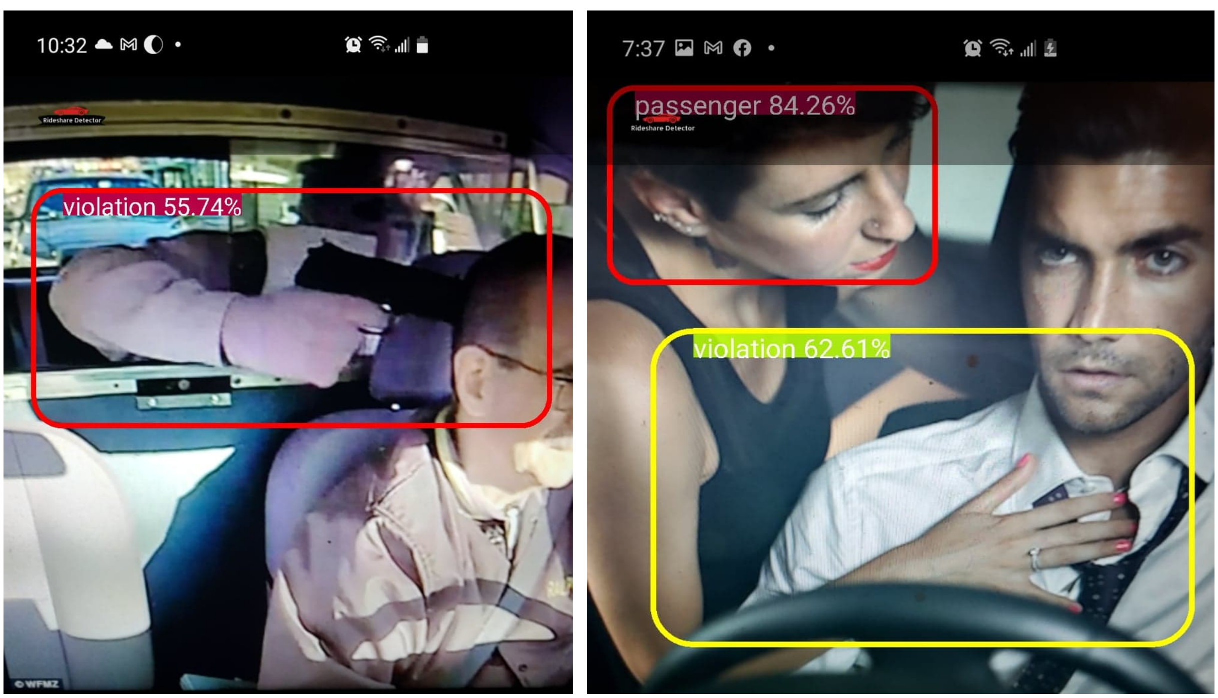

Figure 4.5: Detected Violations a) Violence b) Physical Interaction.

Table 4.4: A Comparison Between SAFEMYRIDES and Other Proposed Systems.

Research	Average Latency	Media type
L. Liu et al. (2018))	1273ms	Sound
(L. Wang et al., 2019)	8345 ms	A 60-second video data
(Long et al., 2017)	2234 ms.	Compressed Videos. Latency (model processing time + network transmission time).
(Ran et al., 2018)	32100 ms.	video chunk is set as S= 1MB, 15 edge nodes
SAFEMYRIDES	620 ms.	Scene view stream

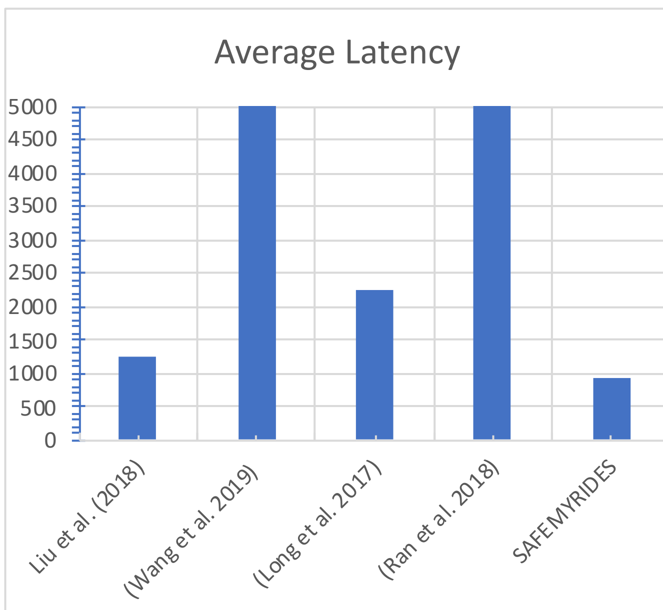

Figure 4.6: Average Latency Versus Previously Developed Systems.

4.11.2 Efficiency

In terms of the resources used by the system, the system used 29.09 MB of the phone's internal storage. The average size of an encrypted media image is 120kB. Energy consumed of running the system for 15 minutes at a rate of 17 m (milliamp), or 79.8 mAH (milliamp hour). Memory used by the system is 104 MB Which means it is consuming less than 1% of the smartphone RAM of 8 GB.

4.11.3 The System Accuracy and Confusion Matrix

In this experiment, we are evaluating the optimized trained network running on the Android S10 plus. The optimized network size is 27 MB, almost 10x less than the actual model size, and runs at 8-bit inference parameters. To get an in-depth analysis of how the optimized network performed, a confusion matrix was created to analyze the accuracy, precision, and recall of the front and rear cameras as in table 4.5 and table 4.6. The confusion matrix was built during a simulated session during the day, trying 51 different normal and violations poses. The front camera and the rear camera were used to detect a violation. The rear cameras of the Samsung S10 plus are 12MP, 12 MP, and 16 MP ultra-wide module cameras. The front-facing cameras are 8MP, 10 MP f/1.9 selfie cameras.

Precision: It tells what fraction of predictions as a positive class were genuinely positive. To calculate precision, use the following formula: $\boldsymbol{TP/(TP+FP)}$ or 0.9111 for the rear camera and 0.906 for the front camera according to the confusion matrix.

Recall: also known as Sensitivity, Probability of Detection. It tells what fraction of all positive samples were correctly predicted as positive by the classifier. To calculate Recall, use the following formula: $\boldsymbol{TP/(TP+FN)}$ or 0.976 for the rear camera and 0.928 for the front camera.
Accuracy: is the measure of what fraction of violation incidences were correctly detected by the system; the accuracy formula is $\text{Accuracy} = (TP+TN)/(P+N)$. Based on the formula, the system's accuracy in detecting correct violation incidences is 46/51=0.9019 for the front camera and 0.862.

Table 4.5: Confusion Matrix of The Proposed System (Rear Camera).

True/Detected	**True**	**False**
True	41 (TP)	1 (FN)
False	4 (FP)	5 (TN)

Table 4.6: Confusion Matrix of the proposed system (Front camera).

True/Detected	True	False
True	39 (TP)	3 (FN)
False	4 (FP)	5 (TN)

4.11.4 Privacy

Surveillance systems have raised apprehension about their threat to individuals' privacy rights (Cheung, Venkatesh, Paruchuri, Zhao, & Nguyen, 2009). Privacy protection is a core objective in the proposed system through ensuring transparency, authorization, and encryption.

Transparency means that both the driver and the passenger are aware that the ridesharing session is monitored. A warning of "Violation Detected" is declared when the system detects a violation. Media surveillance is privacy-intrusive because it allows the observation of certain information that is considered privacy intrusive such as a person's identity or characteristics (age, race, gender) (Senior, 2009) . Therefore, authorization of users is required to monitor ridesharing sessions. Moreover, negotiation of privacy preference (Consortium, 2002) or flexible authorization is offered when users prefer monitoring at night sessions or in remote destinations.

Privacy protection technologies have focused mainly on different visual obfuscation techniques. Encryption of violation incidences is applied to protect sensitive personal information from being saved and transferred securely (Diffie & Landau, 2010) . In the proposed system, an XOR operation with a (2 bit) encryption key is applied for every detected violation incidence and saved in a secured folder in the ridesharing phone. The detected violation incidences are only shared with ridesharing agents or official legal authorities.

4.12 Evaluating SAFEMYRIDES on Different Devices

In this section, the SAFEMYRIDES is applied to a different device to notice the discrepancy in the system performance with different hardware specifications. In the previous section, the system was applied to a Samsung S10 plus device. In this section, the system will be applied to an LG V30 plus phone to compare and contrast the system's performance on different hardware settings.

LG V30 plus has only two rear cameras (16 MP and 12 MP) and one 5MP front camera compared to three rear cameras and two front cameras in the Samsung S 10 plus. The LG V30 Adreno 540 GPU is compared to an Adreno 640 GPU in S10+. Also, the S10 + has a Li-Ion 4100 mAh Battery, and 8 GB RAM is compared to Li-Po 3300 mAh and 4 GB in LG V30. Table 4.7 and Figure 4.7 compares the performance of SAFEMYRIDES on the two devices in terms of average latency, average accuracy, and efficiency (the resources used) for running the system for 15 minutes. The comparison is performed using the rear cameras because of higher resolution and night shots adjustment features.

While the Samsung S10 + achieved higher performance and less latency than The LG V30, the resources consumed (battery and RAM were higher). Samsung S10 + has three cameras which consumed more battery power and RAM in return. There is almost a 4 % difference in the accuracy between the LG V30 and Samsung S10 + and more than 350 ms difference in Latency.

Table 4.7: Performance Comparison for Applying SAFEMYRIDES on Different Devices.

Metric/ Device	LG V30 (Rear Camera)	Samsung S 10 plus (Rear Camera)
Accuracy	0.8845	0.9019
Latency	1297 ms	920 ms.
RAM	89 MB	104 MB
Battery Consumption	48.7 mAH (milliamp hour)	79.8 mAH (milliamp hour).
Internal Storage	29.8 MB	28.09 MB
Average CPU	0.72	0.64

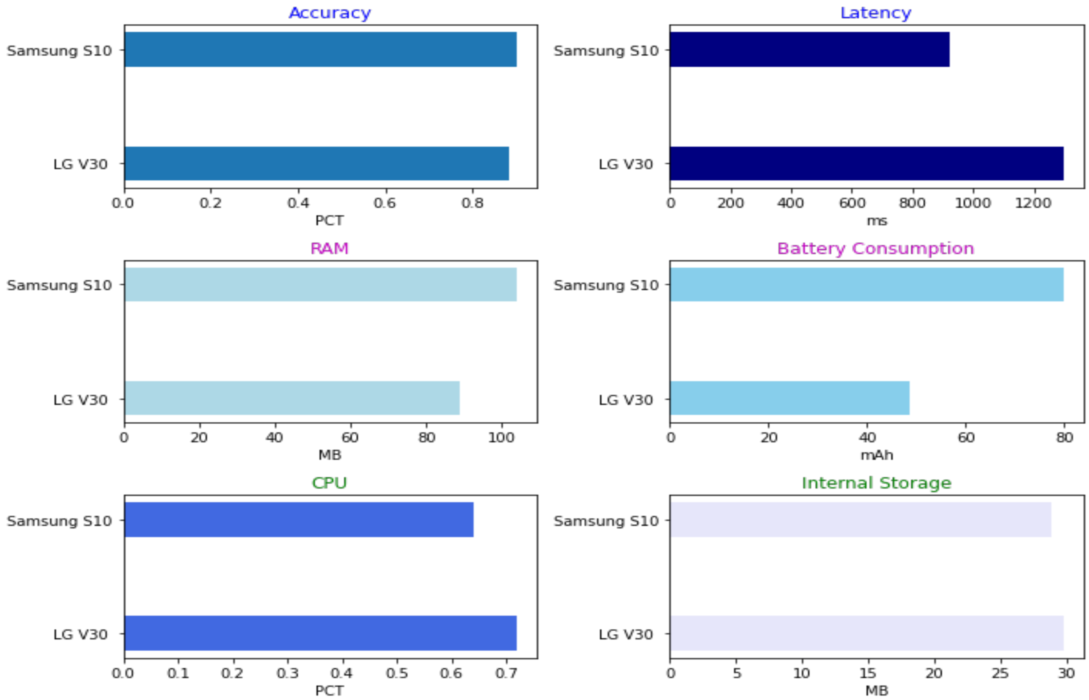

Figure 4.7: Performance Comparison between LG and Samsung using Rear Camera

4.13 Conclusion

The research is introducing a new monitoring system to ridesharing service. The proposed scene-aware system aims to detect violations almost at run time with an average latency of 620 ms. The system uses the smartphone camera scene stream to monitor ridesharing sessions with no actual recording. The system was evaluated technically and functionally in terms of service quality dimensions of *efficiency, accuracy, responsiveness, and privacy.* The system was evaluated using controlled experiments by stimulating different violation poses in real ridesharing settings.

The system was designed to ensure privacy by recording and saving violation incidences in hidden folders in an encrypted format with an average file size of 120KB. Therefore, the private data of the passenger/ driver are secured in the ridesharing device. The system responsiveness was evaluated through monitoring the system average latency or the average time taken to detect a violation. The system achieved the least latency among other developed systems saving the time consumed to transfer images for processing on edge nodes. Instead, the system detects violations locally on the driver's smartphone. Given the limited number of training samples, the system achieved a 90% accuracy rate for detecting 51 different violations poses with a 97% sensitivity rate. We can notice that the accuracy of the full network before optimization was 93.1%. However, this is a typical case in current optimization techniques where there is always a tradeoff between network size and accuracy or between inference latency and accuracy (Kunkel et al., 2019). On the other hand, we can notice the significant difference in inference time (delay) between the full network of 3.56 seconds and the optimized network of 920 ms.

4.13.1 Contributions to Theory

According to Gregor, the theories produced in the research are type I and type V theories or analysis theories and design theories (Gregor, 2006). In Analysis and description theory, an analysis of the phenomena of interest and analysis of the relationships among system constructs are provided. In the design theory, explicit prescriptions (e.g., methods, techniques, principles of form and function are given. The exploration through design is the process of theory building in design science. Theory building aims at solving a real-world problem to achieve practical relevance and developing a theoretical contribution to achieve scientific (Holmström, Ketokivi, & Hameri, 2009). So, “Rather than producing general theoretical knowledge, design scientists produce and apply knowledge of tasks or situations in order to create effective artifacts.” (p. 253) “Rather than posing theories, design scientists strive to create models, methods, and implementations that are innovative and valuable.” (p. 254) (Venable, 2006)

This research contributes to the Information System research by applying the concept of decentralized control where an IoT (smartphone) can independently detect violations and make decisions. The research is one of the early attempts to apply different optimization methods to deep learning networks to run efficiently on an IoT. The research sheds light on the importance of limiting cloud/edge computation and limiting network communications while enhancing the decisions making process and analytics of the IoTs. The reader may recall that in subsection 1.2 we discussed Characteristics of IIS, and in subsection 1.4 we discussed contemporary Challenges of IIS that would be address in this dissertation. In tables 4.8 and 4.9 we outline how the Characteristics and Challenges of IIS are addressed by our SAFEMYRIDES artifact.

Table 4.8: IIS Characteristics and SAFEMYRIDES

Characteristic	**How It Is Addressed by SAFEMYRIDES**
Interactivity	SAFEMYRIDES is a decision-making system. The system asks users whether they would like their sessions to be monitored. The system also warns both the driver and the passenger that a violation is detected and recorded. So, the monitoring process is not passive, but it involves an active agreement and notification of users.
Event Detection	This characteristic address the sensitivity of the system in detecting violations. The system is designed to detect violation throughout the entire ridesharing session. The system showed the ability to detect the violations it was designed to detect in different situations and light conditions.
Communication Skills	The system interacts with the key players of the system. The system interacts with the ridesharing agent by reporting violation incidents in an encrypted format. The system also interacts with drivers and passengers by warning them that a violation is detected. The system communicates with passengers and ask them if they would like to monitor their sessions.
Adaptation	SAFEMYRIDES adapts the change in the environment. For example: the system adapts the change in the surrounding conditions such as changing the detection from day light to night light. It also can detect different types of violations such as physical interaction, violence and seduction.
Predictive Capabilities	The system is designed to predict what object is considered a passenger or driver and what action is considered a violation. The system uses a deep learning model that is trained to detect different types of violations.

Table 4.9: IIS Challenges and SAFEMYRIDES

Challenge	How It Is Addressed by SAFEMYRIDES
Responsiveness	The average system *Latency* is a measure of the system responsiveness. Latency is represented as the average response time to detect a violation. The average response time to detect violations is 620 ms which is the least among current systems
Privacy	Privacy is maintained through transparency where both the driver and the passenger are aware that the ridesharing session is monitored. Also, authorization of users is required to monitor ridesharing sessions. The system is offering flexible authorizations for users who prefer monitoring ridesharing sessions at night or in remote destinations. Private user information is maintained secure by encrypting violation incidences before they are reported to ridesharing agents.
Interactivity Level	The system provides verbal and written interactivity levels. The verbal interactivity is represented when the system voice out, “Violation detected”. The written interactivity level is represented when the system asks the users if they want their sessions to be monitored all the time or at certain circumstances.
Efficiency	In terms of the resources used by the system, the system used only 29.09 MB of the phone's internal storage despite running a deep learning model and the Memory (RAM) usage was just 104 MB despite running real-time inference, which means that the system is consuming less than 1% of the smartphone RAM of 8 GB.
Tangibility	The tangibility dimension represents the visual aspect of the service provided. As displayed in the results figures 4.5, the system runs a camera stream where

	objects and actions are continuously detected. That's mean that the systems add another tangibility dimension to ridesharing services by adding a camera streams that run object detection continuously during the monitoring process

4.13.1 Contributions to Practice

The proposed system is trying to solve the safety issues related to ridesharing, and it has several implications to practice. Firstly, the system allows local ridesharing monitoring on the driver's smartphone, saving the cost of an additional device for monitoring. Having the monitoring system running on the same device used in the ridesharing will ensure efficiency and privacy. Secondly, the system uses no network communication while detecting violations and limits the network communication to reporting the violation incidences to the ridesharing agent. Minimizing network communication ensures privacy, security, and efficiency. So, the system could be used where poor network connections exist. Thirdly, encryption of ridesharing incidences ensures privacy and security. So, the passenger/driver data are only shared directly with the ridesharing agent in case of detected violation.

4.13.2 Limitations and Future Research

When the system was applied to different devices, a difference in performance was detected, affecting the accuracy and latency of detection. The higher is the resolution of the phone's camera, the better is the accuracy. Moreover, the higher is the GPU/CPU, the less is the system latency. So, detection using the rear camera is better than using a front camera. Applying the system on different devices with different specifications, we noticed there is not much difference in detection accuracy during the day, but there is a difference in detection accuracy at night. In general, lower detection accuracy at night calls for increasing the number of night training samples. Moreover, since most smartphone cameras don’t have night vision, the camera ability to balance colors of night images affects detection accuracy. Also, the false-positive rate calls for training the network with more regular ridesharing poses. The system is efficient to use in terms of storage (29 MB) and memory consumption (less than 1% of active memory (RAM) consumption). However, the energy consumption of the battery used by the system is considered above average, but most cameras application are significantly consuming batteries.

The implementation of the system was performed on Android devices only. Other platforms should be tested to compare and contrast the system performance. The size of the dataset used is limited. Increasing the training and validation samples will positively affect the accuracy and the latency of predictions. In addition, the encryption of violation incidences was performed using a simple XOR function. More sophisticated encryption methods using a 128-Bit key are preferred. Also, additional privacy preservation mechanisms such as video inpainting and data-hiding schemes will add extra *confirmation* on user privacy (Cheung et al., 2009). Since the system achieved lower accuracy and decreased sensitivity in detecting violations during the night, the deep learning network needs to be trained with more night samples, where poor light obstructs definite detection of violations.

Further research on the optimization of camera usage is required in terms of camera battery consumption assuming that a ridesharing driver will be using the system for hours. Moreover, the system should be deployed in a real setting to address other intangibles service quality measures such as reliability, understanding, and courtesy.

Other future work includes implementing the system on other platforms such as IOS and Raspberry PI. The system should be tested on different platforms to compare other performance measures such as stability, availability, and convenience. The system used compression and quantization techniques as optimization methods for the Mobile SSD. However, applying other optimization techniques such as pruning might enhance the optimized network accuracy. Moreover, benchmarking for the proposed system is required because the detection accuracy differs between different phone cameras. Benchmarking will help to determine the minimum required specifications to obtain reasonable detection accuracy.

Chapter 5 A Multi-Criteria Decision Making for Real Estate Based on Degree of Damage: Towards Enhanced Service Quality.

5.1 Introduction

The real estate industry is interconnected to many vital sectors of the economy, including the finance, insurance, and engineering sectors. Real estate investments involve many social, governmental, environmental, and financial factors (Del Giudice, De Paola, & Forte, 2017). Existing work on real estate investment focuses predominantly on the trend predictions of house pricing exclusively from structural and financial factors (E. S. Kumar, Talasila, Rishe, Kumar, & Iyengar, 2019). However, little attention was given to enhancing the real-estate search process (Jarosz et al., 2020).

The service quality of real-estate websites has been affected by pre-narrowed search tools and insufficient knowledge support for customer decision-making (Yuan, Lee, Kim, & Kim, 2013). According to Trulia (J. Chen, Hui, & Wang, 2011), 44% of real estate consumers (almost one in two) regret their purchase or rent decision. The leading causes of these regrets are a lack of comprehensive information about properties. Such regrets can be eliminated or reduced by enhancing the *information quality* and *relevancy* of search results (Ullah, Sepasgozar, & Wang, 2018).

Research on real estate service quality revealed that images were considered the most important feature on real-estate websites (Choi & Kim, 2017) (Ullah & Sepasgozar, 2019). Real estate images incorporate critical structures and semantic features such as the degree of damage (Ciuna, Salvo, & Simonotti, 2017). The degree of damage is a sematic feature representing the degree of physical deterioration of the building sections represented in ordinal scale, i.e. (1 extreme damage and 3 is minor damage) (Ciuna, Salvo, et al., 2017). Information extracted from images is considered one of the "*Tangibles*," the visible aspects of the service quality, to improve customer satisfaction (Panda & Das, 2014). *Tangibility* is a primary *service quality* dimension where the service representation gives a clear, concrete image of the service (Santos, 2002). Including the degree of damage in the real estate search process would enhance *tangibility,* information quality, and hence *service quality*.

The second most important feature of the website is the 'tools' as search functions. (Ullah & Sepasgozar, 2019) found that search tools raise users' self-efficacy by providing users the flexibility to search based on their preferences. Automatic real estate agents such as Realtor and Trulia offer limited spatial features that hinder the accessibility to information to reach *relevant* and *precise* results (Samaa Elnagar & Thomas, 2019) (Xu, Benbasat, & Cenfetelli, 2013). Moreover, the *accuracy* of the real estate searching process is strongly correlated with the *relevancy* of search results to user interests (Gargallo Valero, Salvador Figueras, & Miguel Álvarez, 2017; Santos, 2002). The real estate search process may produce many relevant tradeoffs, but traditional real estate search methods are single-faceted, thus inhibiting customers from comparing properties from many angles. So, the limited real estate filters have put a burden on users matching their criteria of preference or capture the key *relevant* information (Ciuna, De Ruggiero, Manganelli, Salvo, & Simonotti, 2017; Hallerbach & Spronk, 2002) (Jagadeesh, Piramuthu, Bhardwaj, Di, & Sundaresan, 2014).

The multi-dimensional nature of real estate search decision problems inherently lends itself to be a multi-criteria decision problem (Rabiei-Dastjerdi, McArdle, Matthews, & Keenan, 2021). Applying multi-criteria decision-making (MCDM) technologies allow users to prioritize their criteria of preferences. MCDM ranks properties to match user preferences, so users browse less to reach desired listings, and therefore MCDM enhances the *efficiency* of the search process (Brunelli, 2014). The MCDM would offer *flexible* search criteria to match user preferences (Chimbalkar, Patil, Jain, Kate, & Dwivedi, 2014).

The research introduces a new explanatory real estate filter "degree of damage" that filters properties based on the state of maintenance. The degree of damage is extracted automatically from real estate images using the deep learning network of Mask Region-based Convolutional Neural Network (Mask-RCNN) (He, Gkioxari, Dollár, & Girshick, 2017; Nie, Duan, Ding, Hu, & Wong, 2020). The research adds more expressive filters that boost the current real estate website's *service quality* and outcomes in terms of *efficiency*, *flexibility*, *relevancy*, and *tangibility*. So, the two main contributions of the system are:

a. The first is to add a micro-level filter that allows filtering real estate by their degrees of damage or their states of maintenance. This filter is expected to add to the real estate websites' service quality *tangibility* and avoid inaccuracies caused by overlooking such a key filter.
b. The second is to create a macro-level MCDM system that incorporates different real estate filtering criteria. The system enhances the quality of the real estate searching process in terms of *relevancy*, *efficiency,* and *flexibility*.

The system would have many social impacts, such as improving customer satisfaction in electronic real estate agents. Based on the above discussion, we could formulate the research questions to

a. How could the proposed system enhance the *relevancy*, *efficiency,* and *flexibility* of the real estate search process?
b. How could introducing the degree of damage feature enhance the user's satisfaction through enhancing the service quality *tangibility* and search results relevancy?

The following sections review prior research on real estate search and filtering systems and the literature on real estate service quality. Then, an overview of the degree of damage feature detection is given, followed by an overview of MCDM and AHP. Afterward, we present the methodology of the proposed system. The architecture of the system is described and evaluated experimentally in real-life simulations. Finally, there are future work and limitations.

5.2 Literature Review

5.2.1 Service Quality and Information Quality Issues in Real Estate

(Choi & Kim, 2017) found that providing high-quality search filtering service is required to improve the information quality and provide users with quick and reliable information. An improved search filter service that satisfies user's needs would enhance satisfaction in online real estate services. Ullah and Sepasgozar (2019) developed RESTAM model using the key constructs of TAM (technology acceptance model). The framework found the significant impact of search tools development on *Information Quality*, *Service Quality,* and *System quality*, which corresponds to the importance of search tools optimization for the up gradation of online real estate.

(Yuan et al., 2013) found that the use of the Internet does not benefit homebuyers in terms of search time and flexibility. While it does encourage buyers to search more intensively, it also wastes more time and energy browsing irrelevant results. Studies also found a strong correlation between the adoption of AI techniques and the increase in understanding the real estate markets in addition to user satisfaction and trust (Winson-Geideman & Krause, 2016).

(Ullah & Sepasgozar, 2019) confirmed the lack of the dynamics that govern the adoption of information technologies in the real-estate business that affected system quality, information quality and service quality. (N. Wu, Gelman, & Osesina, 2009) found potential data quality problems in real estate websites such as accuracy, timeliness, believability, relevancy. These problems can impair the user's decision-making and prevent the effective use of information. Potential problems arise when data on the top-quality ranking websites are insufficient hindering data of great importance to the user's decision making.

5.2.2 Previously Developed Systems

(Athalye, 2013) developed a real estate recommender system that combines the characteristics of both *Content-Based* and *Collaborative centered filtering* methods, considering the advantages of this hybrid method to rectify the limitations of both methods and deliver proactive references to a newsreader. However, the research didn't include any semantic features, such as the degree of damage in the recommendation process. Ng, She, Cheung, and Cebulla (2016) have implemented a vacation rental portal to improve penetrating efficiency. The research used three techniques: images-based cosine similarity calculation, textual explanation established Jaccard similarity calculation. The travel accommodations recommendations were based exclusively on image comparisons or textual description-based similarities. However, traveling choices of users could not be seized entirely using either methods. Running cosine similarity on similar rental property images ignores other geographic and financial features that affect the rental process.

(Samah et al., 2019) designed and developed a web-based house purchase recommendation system (HPRS) using a Genetic Algorithm (GA) approach to overcome the limited filtering features that consider user's preferences. However, applying genetic algorithms might be slow and complex to apply to an active search system. Gargallo Valero et al. (2017) and McCord, McCord, Davis, Haran, and Bidanset (2019) used the eigenvector spatial filtering (ESF) to analyze the local variation and spatial heterogeneity. ESF are incorporated as predictors markers for exclusion or inclusion criteria. However, the research focused primarily on spatial features, which are insufficient for a comprehensive real estate purchase decision-making.

Several online APIs can evaluate properties conditions, such as restb.ai [5] However, these APIs cannot provide criteria that explains the evaluation process nor allocate the damage in real estate components. Missing clear criteria for evaluating the damage resulted in highly inaccurate evaluation. For example, a living room image with colossal damage in the ceiling might be detected as an average condition. Localizing and quantifying each image will help users understand the rationale behind assigning certain condition to the entire property.

Very little research tried to estimate the visual damage conditions of real estate (Poursaeed, Matera, & Belongie, 2018) developed a deep learning system to detect the luxury level of a property using the interiors and exteriors images. The system uses the luxury level along with other parameters for the value assessment process. The authors used DenseNet networks to estimate the luxury level of each property section. However, the luxury level of the house is not sufficient to evaluate the actual condition of a house. Samaa Elnagar and Thomas (2019) used Mask R-CNN to estimate properties damage conditions integrated with Automated Valuation models (AVM) for property price estimation. However, the system targeted real-estate agents to enhance the AVMs' performance without allowing customers to filter real estate based on damage conditions.

One of the most popular filtering methods used in matching real estate users' preferences is *Collaborative filtering* which gathers user interests and matches them with alternative users' preferences or ratings for things (collaborating). The fundamental theory of the cooperative filtering approach is that if a user one has the similar style as a user a pair of on associate item, user one presumably has user 2's opinion on a unique item x than to has the style on x of an

[5] https://restb.ai/solutions/property-condition/

individual chosen haphazardly (Schreyer, 2018). Assuming the correlations between different users' behavior in the case of real estate is highly insignificant.

Another popular approach is the *Content-based filtering* approach that makes recommendations by analyzing the description of the items using keywords. The similarity of products is calculated based on the associated features of the items (Satapathy, Jhaveri, Khanna, & Dwivedi, 2020). However, the model performs well using hand-engineered features. Given that current real estate websites provide limited filters, the model can only make recommendations based on the existing filters, such as location, area, and year built. Other vital filters such as schools, crime rate, and degree of damage are excluded. So, the model can only recommend based on existing filters (Badriyah, Azvy, Yuwono, & Syarif, 2018).

Thus, real estate websites users may not always express their preferences in a manner that easily matches their requirements due to limited search tools and filtering methods (Acharya, Kagan, & Zimmerman, 2010). The gap in literature could be summarized as the lack of explanatory real estate filters that led to a deficiency in the service quality of real-estate websites (Ullah & Sepasgozar, 2019). McCluskey et al. (2013) remarked that the relationship between property value and its explanatory filters is highly complex and generally non-linear, which calls for more insightful filters than the traditional spatial features that failed to fully represent a property. However, there is an agreement within real estate applications that more advanced features and search tools to enhance the efficiency and relevancy of the search results needed for informed real estate decisions (Limsombunchai, 2004).

5.3 Conceptual Framework

The *information quality* and *service outcome* directly impact real estate websites' *service quality* and *customer satisfaction* (Dabholkar & Overby, 2005; DeLone & McLean, 2003) (L. Zhao et al., 2012). *Service quality* in real estate might negatively impact user perception because of the lack of detailed property photos and neighborhood insights (Ali et al., 2021; Dabholkar, Shepherd, & Thorpe, 2000; Ullah et al., 2018).

The introduction of new technologies such as virtual showcasing and AI has a positive correlation with service quality (Blayse & Manley, 2004). In this research, deep learning is used to identify and localize the degree of damage in real estate images to improve service quality tangibles. Nonetheless, a good perception will develop when users find the information *relevant* to make *efficient* knowledge-based informed decisions (James III, 2009; Xu et al., 2013). The lack of relevant information happens when users find desired listings after browsing several irrelevant listings. The user uses limited 'tools' as search functions that display listings to match search results nut many of which are irrelevant listings. (Ullah & Sepasgozar, 2019) found flexible search tools raise users' self-efficacy by providing users the search filters that satisfy their requirements.

Introducing MCDM will help match user preferences and provide a *flexible* search service where users can choose and elucidate their preferences as part of a decision-making process. MCDM approaches have successful contributions to the quality of financial-related decisions such as real estate purchases (Abdel-Basset, Mohamed, Sallam, & Elhoseny, 2020; Zopounidis & Doumpos, 2002). MCDM ranks properties according to user preferences so users can find relevant listings first. Therefore, MCDM improves not only the *relevancy* but also the *flexibility* of the search process (Hallerbach & Spronk, 2002). Accumulating the issues addressed in the literature with service quality models of real estate service, we can summarize the main service quality issues that are addressed in the research context as in Figure 5.1.

Efficiency: it directly impacts both *service outcome* and *information quality* (L. Zhao et al., 2012). *Efficiency* means the reduction of search costs, the availability of information, and the possibility of multi-attribute comparisons (Xue, Harker, & Heim, 2000). In this research, the efficiency is measured by difference in the number of browsed listings using the proposed system and the conventional real estate websites to find the desired listings.

Flexibility: defined as the freedom that online environment offers to ease user navigation through websites (Alba et al., 1997). Compared with current real state websites, which offer a limited number of search criteria, the research offers a flexible selection pool of property criteria and allows prioritizing one criterion over the other. So, we express *flexibility* as the number of criteria the users use to prioritize their preferences in comparison to those offered in popular real estate websites. *Flexibility* always coincides with *Interactivity* (Yadav & Varadarajan, 2005). So, supporting websites' *flexibility* will increase interactivity and consequently better service quality (Pur, 2017) (Jiang et al., 2010).

Relevancy: search relevance is the accuracy of the relationship between the search query and the search results (Goldman, 2005). *Relevancy* is affecting both the *information quality* and *service quality* of real estate websites. In the proposed system, we measure relevancy by the accuracy of the MCDM in ranking listing according to user elucidated criteria.

Tangibility: is the clear reflectiveness of resources necessary for providing services to customers (Pakurár et al., 2019) (Sun et al., 2012). The degree of damage answers a significant quality problem raised by Sun et al. (2012) "Are the pictures of each property showing enough details? Is sufficient information provided for each property?" The degree of damage localizes damage in each property image and adds necessary information to real estate decisions. So, we measure *tangibility* by enhancing the information representation through adding localized damages to real estate images (Seiler & Reisenwitz, 2010).

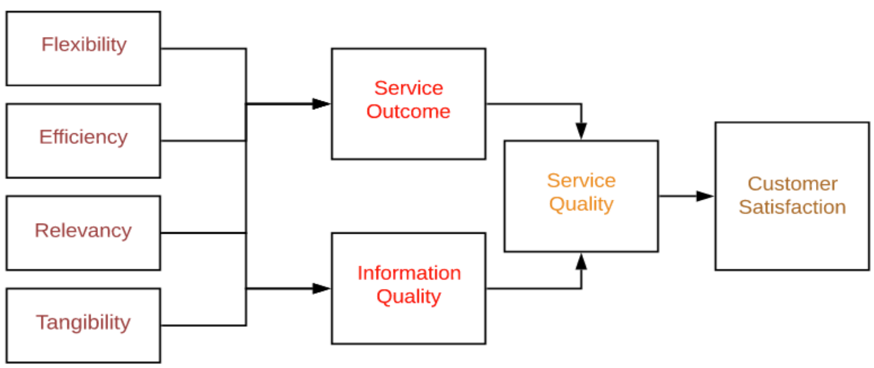

Figure 5.1: Research Approach to Enhance Real Estate Service Quality.

5.3.1 Degree of Damage Filter Motivations

The current challenges in the real estate filtering process inspired introducing the degree of damage to filter real estate. Real estate images include rich semantic information about a property. With the advent of deep learning for computer visions, automatic damage detection of real state images has been an attainable, realistic task. The motives to introduce the degree of damage filter are:

5.3.2 Cut Costs of Initial Inspection and Avoid Human Judgment

Determining the degree of damage by experts is a costly subjective process (d'Amato, 2018). On-site manual property valuation is time-consuming and based on subjective judgments. Sometimes, the assessment is based on the property's negotiated price rather than estimating the true market value (Mousa & Saadeh, 2010). Moreover, a user's tolerance to a property's degree of damage differs widely based on user interests. An investor might have a greater damage tolerance than a first-time homebuyer. Providing an initial estimation of real estate damage conditions would save the user time and cost in finding relevant search results (You, Pang, Cao, & Luo, 2017).

5.3.3 Much Focus on Spatial Filters

Common real estate filters include price range, no. of bedrooms, Sqft, and zip codes. Those filters represent a collection of spatial layers of individual transactions (Limsombunchai, 2004). The lack of environmental and semantic attributes, such as the degree of damage, has led to heteroscedasticity between house attributes and house price leading to customer dissatisfaction (Murakami & Griffith, 2019).

5.4 Methodology

This research is built upon Samaa Elnagar and Thomas (2019), which used Mask R-CNN for image-based real estate appraisal. The Mask R-CNN extends the Faster R-CNN deep learning network that uses instance segmentation instead of bounding boxes (He et al., 2017). Instance segmentation allows precise pixel-wise localization of objects through an attached binary mask of the detected object, especially if these objects are in small heterogeneous shapes such as damages and cracks. Mask R-CNN has been used successfully in many medical and industrial applications (Lu, Kong, & Guan, 2020).

In this research, we are following a design science approach in developing the decision support system and the novel degree of damage filter (A. R. Hevner et al., 2004), which are considered design science artifacts (Peffers et al., 2012). The Degree of Damage filter is an instantiation, and the entire system is considered a framework. Both artifacts are implemented and instantiated.

Prototypes are used to evaluate the degree of damage, and the whole system is evaluated using *illustrative scenarios* in synthetic or real-world situations to demonstrate their utility.

5.4.1 Degree of Damage

The degree of damage is a composite weighted attribute. The attribute is inspired by the State of maintenance (STM) metric developed by Ciuna, Salvo, et al. (2017). STM represents the degree of physical deterioration of a building. STM is divided into internal and external sections. Each section has its significance weight, which is determined by real estate experts, depending on its susceptibility to damage, as shown in Figure 5.2. For example, bathrooms on the upper floors are more prone to floor damage and leak issues than bathrooms on the ground level. Thus, they have higher significance weights. Each section consists of components, and each component is giving an importance weight based on its cost of maintenance. For example: the ceilings are usually given higher weight than the walls because of their maintenance cost. There are four levels of damage in the ordinal scale: 1 Extreme damage or poor state of repair, 2 Moderate Damage or insufficient state of repair, 3 Minor Damage or tolerable condition of maintenance, 4 None or good state of repair. However, if a component, e.g., the ceiling has more than one damage of different degrees. The component is assigned the least degree of damage (or highest damage level). For example: if a ceiling has two minor damages and one extreme damage. The ceiling is assigned an extreme degree. The section degree of damage is calculated as:

$$Sec_{li} = \sum_{j=1}^{n} W_j \, max\,(\,CD_j\,) \qquad (1)$$

Where Sec_{li} is the interior or exterior l section. The section has n number of components, and each component j is given a weight W_j based on importance. The component degree of damage is CD_j.

Figure 5.2: Degree of Damage in Different Sections and Components.

5.4.2 Multi-Criteria Decision Making

In searching for real estate, a single criterion is insufficient to assess a set of available options. For example: Although location is an important criterion, it is not a sufficient filtering criterion as long as other vital criteria are not included, such as schools, crime, and association fees (HOA). Multi-criteria decision-making (MCDM) is a field of operational research wherein the decision alternatives are analyzed with respect to multiple (and often conflicting) criteria (Ishizaka & Siraj, 2018). MCDM assists in framing decision problems, exploring tradeoffs, and raking different filtering criteria to match user preferences (Aronson et al., 2005; Osei-Bryson, 2004).

There are several MCDM methods. The most preferred models are the *utility-based models,* including *MAUT, AHP, Weighted Sum Method, and Weighted Product Method* (A. Kumar et al., 2017; Sugumaran, 1998). AHP simplicity has gained popularity, especially with comprehensible MCDM tasks (Figueira, Greco, Roy, & Słowiński, 2013). The logic behind AHP is to rank different criteria used to evaluate different alternatives. The core of AHP is quantifying the weights of the decision criteria through the comparison of criteria pairs instead of sorting, ranking, them (LeCun et al., 2015a). The procedures for performing AHP in real estate filtering are:

5.4.3 Define Problem and Criteria

Using AHP enables the users to customize the filters used in the real estate search process. The alternatives are the available real estate available on different websites. The criteria include but are not limited to property condition, location, schools, crime rate, and the number of rooms. AHP here allows specifying a range for each criterion rather than ranking them. AHP first decomposes the decision problem into a hierarchy of more easily comprehended problems, each of which is analyzed independently.

5.4.4 Establish Priorities in a Comparison Matrix

Once criteria are well defined, a pair-wise comparison is performed. The comparison between two criteria is given on a scale from one to nine, where nine represents an extreme preference for one alternative over the other. The comparisons are gathered in a comparison matrix $\boldsymbol{A}$. The eigenvalue method then uses the comparison matrix $\boldsymbol{A}$ to calculate the local priorities or weights as in equation 2, where $\vec{p}$ is the priorities/weight vector, $\lambda_{max}$is the maximal eigenvalue.

$$A \cdot \vec{p} = \lambda_{max} \cdot \vec{p} \qquad (2)$$

5.4.5 Calculate the Consistency Index

After removing redundancy from the comparison matrix, the consistency index (CI) is calculated to verify user consistency in the choices between pairs of criteria. For example, if the user extremely prefers the location over the property age, the user should consistently prefer property age less than the location. For such a matrix, ω is the eigenvector (of order n), λ is the eigenvalue, and the CI is:

$$\text{CI} = \lambda_{max} - n/(n-1) \qquad (3)$$

5.5 The Proposed System

The architecture of the proposed system is shown in Figure 5.3. There are three main modules in the system: damage detection, degree of damage estimation, and the search and retrieval system. Each module is discussed below in detail:

5.5.1 Property Damage Detection

This module is responsible for detecting damage in each property image. There are three cases where the damage detection module will activate.

1- If a realtor submitted new property images to the system, the damage detection module automatically run damage detection on each of the property images if the computing units are not busy.
2- If the computing units are busy, unidentified properties will be scheduled for damage detection when the system is idle.
3- If a user submits a search/filtering request and some retrieved properties are not damage detected, they will be given the highest priority to assign a degree of damage.

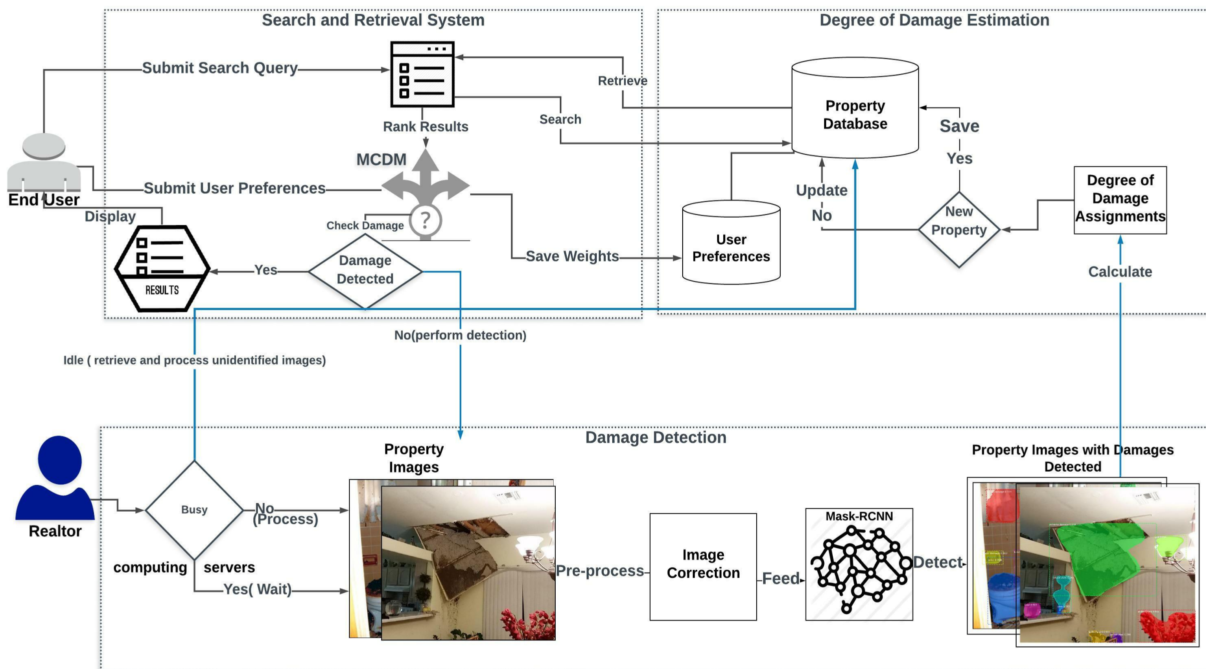

Figure 5.3: The Proposed System Architecture.

5.5.2 Image Correction

Each image goes through essential image corrections as basic preprocessing steps. Image correction includes blur, contrast, brightness adjustment, rotation, and skew correction. In addition, the system rejects images with low resolutions or pixel dimensions less than 72 PPI. Corrected images are now ready to be fed to the Mask-RCNN network.

5.5.3 The Mask-RCNN Network

A trained Mask-RCNN network is employed to detect damage in different sections and components in real estate images. The Mask-RCNN consists of a backbone network and a head network. The main component in the backbone network is the faster R-CNN. The head network is responsible for detection and object annotation. For feature extraction, the Feature Pyramid Network (FPN) is used to build a bottom-up pyramid-like feature architecture (Lin et al., 2017). FPN learns different degrees of damage features in fine granulation order, starting from coarse structure features to fine features such as colors and surface areas.

5.5.4 Property Images with Damage Detected

The output of the Mask-RCNN network is the damage detected real estate images where damage is localized along with surrounding bounding boxes. The network also gives a confidence score for each detected image detected, as shown in Figure 5.4. The confidence score represents how confident is the network about assigning a specific class to a detected object (Z. Huang et al., 2019).

5.6 Degree of Damage Estimation

This module has two components. The first is the *Degree of Damage assignment,* which assigns a degree of damage to the entire property based on damages detected in each image. The second is the property database, where properties are saved while updating their assigned degree of damage.

Figure 5.4: Damage Localization along with the Surrounding Bounding Box and Confidence Score.

5.6.1 Property Database

To gain a deeper understanding of how the data flow in the proposed system and how the Degree of Damage will be reflected in the real estate data records, the system data model is expressed in Figure 5.5 and explained below in detail.

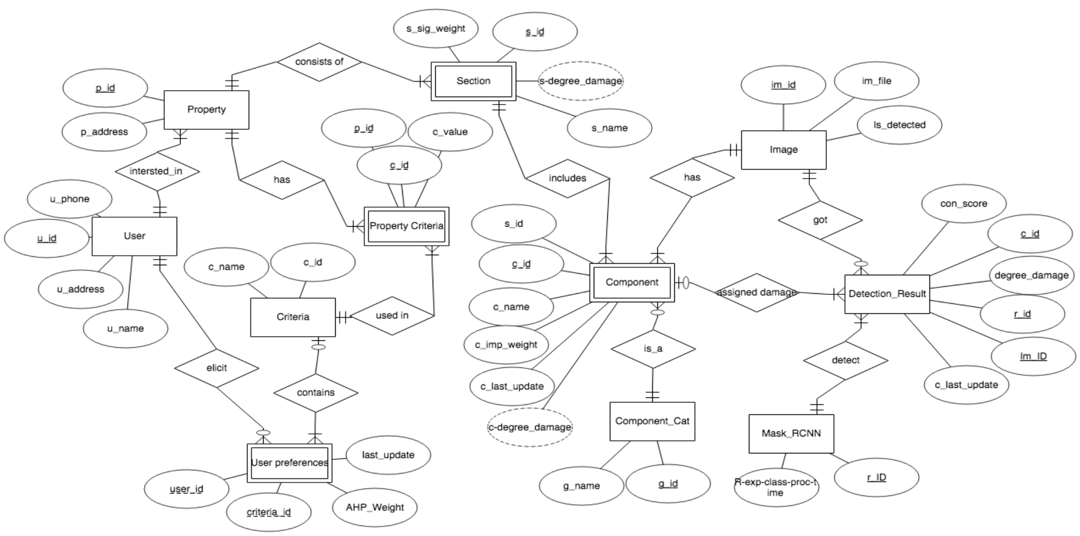

Figure 5.5: The Data Model for the Real Estate Search System.

1. **Property:** each property has an id, address and a set of different criteria pulled from the **Criteria** entity.
2. **Criteria:** it is a lookup table that contains a set of property's criteria such as *no_floors*, *price,* and *sqft*.
3. **Property Criteria:** contains the values for each of the property's criteria.
4. **User:** this entity contains the information of the users, such as email, phone, etc.
5. **User Preferences:** stores the user's preference weights of each criterion in comparison to other criteria. The *AHP_Weight* is the weight generated from the AHP for each criterion.
6. **Section:** or property's sections such as bedrooms, living rooms; the section is dependent on property.
   - *s_sig_weight*: The significance score of the section based on its susceptibility to damage. The values of *s_ degree_damage* is derived from the sum of the damage of its components.

7. **Component:** is each part of a section such as the *Ceiling, Floor, Door, Closet* of a section; component is dependent on section (relative to the section).
   - *c_imp_weight*: real estate experts assign a relative weight to each component, usually based on its cost of maintenance.
   - *c_degree_damge*: is the value assigned by the last damage estimation, and it would be Null if the component isn't previously estimated.
   - *c_Last_Update*: is the date of the last damage estimation or Null if it is not damage estimated.
8. **Comp_Category:** is the components lookup (e.g., Ceiling, Floor, Door, Closet, etc.). This entity is independent of any particular component.
9. **Mask-RCNN_Model:** stores the configurations of the Mask-RCNN models running on different computing units.
   - *R_Exp_Class_Proc_Time*: is the average time spent in processing each of the four damage classes.
10. **Image:** each image contains one or more components of a section. Each image has a name and file location. *Is_detected* marks undetected images to be detected when the Mask-RCNN_Models are idle.
11. **Detection_Result:** for each detected component in the image, the *degree_damge* and *con_score* are saved. *con_score* is the confidence score that Mask-RCNN provides for each detected component.

5.6.2 Degree of Damage Assignment

In this module, the assignment is firstly performed for each component, then section, then for the entire property. Each section represented in the image is assigned a degree of damage based on the summation of its components damage as in equation 1. The total damage TD in a property is simply the average of damages in property sections represented in the following equation.

$$TD = \sum_{l=1}^{n} \sum_{i=1}^{m} W_{li}\ Sec_{li} \Big/ (n.m) \qquad (4)$$

Where n is the number of different real estate categories such as the interior and the exterior, m is representing the number of sections in each category, and W_{li} is the significance weight of each section. W_{li} is usually assigned by real estate experts based on each section's significance as discussed earlier.

The Degree of Damage Assignment activates in two cases: the first is when new property's images are damages are detected. The second is when search results contain undetected properties. The entire property is assigned a damage degree to ease the search and retrieval process.

Usually, real estate websites receive hundreds of listings. Applying deep learning tasks, including training and inference, is burdensome and can result in inefficiencies such as large scheduling overhead (Q. Zhang, Zha, Wan, & Cheng, 2019). Therefore, many algorithms were developed to schedule and predict the completion time of job batches. These algorithms reasonably choose the batch size of the following batch jobs according to the GPUs memory to achieve minimum job latency (G. Chen, Zhao, Shen, & Zhou, 2017).

Examples of deep learning task scheduling are *Nimble* and *EffiSha* (G. Chen et al., 2017; Kwon, Yu, Jeong, & Chun, 2020), which use parallel GPU tasks scheduling algorithms such as *ahead-of-time (AoT)* to minimize the overhead avoid latency. However, every algorithm impacts the performance differently based on the hardware specifications, computation frameworks, the number of GPUs used, and the DL network size. For example: *Nimble* shows different inference times for different networks. In addition, the data return time is relatively irregular (Amaral, Polo, Carrera, Seelam, & Steinder, 2017). Therefore, addressing workload (damage inference) scheduling is out of the scope of the paper. The following algorithm summarizes the property damage assignment process:

Algorithm 1: Degree of Damage Assignment to Real Estate Images.	
Step	**Description**
1- Check if damage detection is required	a) Determine whether up-to-date damage estimate scores exist in the Detection_Result table. b) If the given scores exist, go to step 4; otherwise, go to steps 2 and 3.
2- Schedule images for detection	Undetected property images are assigned for task scheduling.
3- Process each *image*	Each image contains one or more components. Moreover, many images could represent the same section. The Mask -RCNN_*model* will detect damage in each section's component CD_j to be saved as *c_ degree_damage*.
4- Compute *Section* score, s_*degree_damage*	Calculate the section s_*degree_damage* using the *c_ degree_damage* (CD_j) & *c_imp_weight* (W_j) of its *components* as in equation 1.
5- Compute the property score TD	Repeat process 4 for each property section. Then, calculate the total damage TD by calculating the weighted average degree of damage of each property section s_*degree_damage* (Sec_{li}) and s_*sig_weight* (W_{li}) as in equation 4. TD is saved as one of the property's criteria in the *Property Criteria* entity.

5.7 The Search and Retrieval System

This module represents the interface between the system website and the users where the system is responsible for receiving users' search and filtering queries and applying MCDM for the retrieved results from the property database.

At first, the users specify the range values, locations of interest, price range, and area. The results are ordered according to these criteria to downsize properties of interest. Then, the user performs criteria elicitations by prioritizing criteria of interest. Criteria selection pool includes but not limited to *Price*, *Space*, *Age*, *Location, Crime rate, Schools, Degree of Damage, Tax.* These criteria represent common properties attributes based on user interests. However, the system is flexible in adding criteria to the selection pool which are saved as property attributes in the *Property_Criteria* table.

The system highly recommends that the user select the top three criteria of preferences to limit the number of pair-wise comparisons. Still, the system allows selecting more than three criteria, but the number of pair-wise comparisons will be overwhelming for the users to complete. In case of incomplete or inconsistent pair-wise comparison matrix, the system applies one of the incomplete pair-wise solutions such as the Eigenvalue Method (EM) and logarithmic least-squares (LLS). These methods can infer and synthesize missing values using the knowledge of the existing values (Harker, 1987; Tekile, Fedrizzi, & Brunelli, 2021). Then, the system saves user's criteria of preferences to the *User preferences* table.

Secondly, for the top ten properties, AHP performs pair-wise comparisons on each pair of properties for each criterion. Then, order properties descendingly based on their o*verall local score* or the sum of criteria weights. Algorithm 2 explains the search and retrieval process.

Algorithm 2: Applying MCDM to Real Estate Search Results.	
Phase	**Description**
Process Property Database	a) Specify the range values for *Properties* in terms of *Price, Age & Location.* then, order results accordingly. b) for images with undetected damages, apply the ***Property Damage Estimator***. Then, save the results to *Property_Damage_Results* entities.
User Preference Elicitation	a) the user elicits criteria of preference from a set of available evaluation criteria: o *Price*, *Space*, *Age*, *Location, Crime rate, Schools, Tax, Degree of Damage.* b) ask the user to prioritize each criterion with respect to other criteria through pair-wise comparison. However, the default setting is to select three criteria of interest.
Complete, store comparison matrix	a) if the user didn't complete the pair-wise comparison, apply the incomplete pair-wise inference algorithm. b) prioritized criteria and their weights are stored in the *User_Preferences* table.
Apply AHP	a) for each pair of the top ten property results: o apply AHP for each evaluation criteria based on user preferences. b) sum the weights of criteria to compute the *overall local score* for each property
Ranking	a) rank the properties in descending order based on the o*verall local score* and finally display them to the user.

5.8 Implementation

The implementation of the proposed system could be divided into two sections, the first is the implementation of the Mask-RCNN degree of damage detection network, which include training and testing the network with damage images. The trained network assigns a degree of damage to each damage in the image. If multiple damages detected, a component is assigned the highest degree of damage. The second section is responsible for implementing the search/filtering system where MCDM (AHP) is applied to rank the search/retrieval results that match the user preferences.

5.8.1 Implementing the Mask-RCNN Degree of Damage Detector

In this section, the entire process of training the Mask-RCNN degree of damage detection network is explained in detail in terms of data selection, network configuration, defining criteria for determining each degree of damage, the annotation of training and validation images, the training steps, and results.

5.8.2 Data Selection and Image Preprocessing

Deep learning networks usually need vast number of training and validation samples. However, there is no available dataset for detecting damage purposes. So, the training and testing images were downloaded manually from the web and relevant websites.

Due to our limited computational resources, we trained the network to detect damage in a single real estate component which is the ceiling. We downloaded images with different degrees of ceiling damage. The total number of images is 594 images with varying degrees of damage. We selected images of good quality with observed damages. We tried to avoid occupied ceilings with irrelevant objects such as speakers, alarms, and routers. Image preprocessing included *resizing* pictures to 712 x 712, and applying *contrast stretching* to adjust the contrast of images (Munteanu & Lazarescu, 1999).

5.8.3 Degree of Damage Criteria

It is challenging task to quantify the degree of damage of each property component. There are usually some confusing images that can fit in more than one degree of damage. So, we developed criteria based on which we can identify different ceiling degrees of damage:

Minor Damage Class: it represents the small water stains which are beige to light brown in color, small paint cracks (uneven surface), small moldy spots (usually dark in color).

Moderate Damage Class: it represents large water stains, medium size paint cracks, or moldy spots.

Extreme Damage Class: it represents apparent damage in the ceiling gypsum board, large moldy areas, falling parts of the ceiling.

No Damage Class: it represents standard ceilings with no damage, mold, or stains. However, training images are without fancy ceiling shapes that might wrongly be detected as damage.

5.8.4 Image Annotation

We used VGG annotation software[6] to annotate training images with different damage degrees. Each damage is surrounded with a polygon which was represented as a set of points of (x, y) values. Then, each polygon was assigned a class. The annotation file was exported as a JSON file and each image was represented as class in the following format:

[“images.jpg4132”:[“filename”:”images.jpg”,”size”:4132,”regions”:[[“shape_attributes”:[“name”:”polygon”,”all_points_x”:[51,128,231,196,139],”all_points_y”:[84,18,80,106,112]],”region_attributes”:[“name”:”extremedamage”]]]

5.8.5 Training and Testing

An essential step in training any deep learning network is to specify the images used for training and testing. We planned to slice the downloaded images into 30% Extreme; 30% Moderate; 30% Minor; 10% No damage for both training and testing. However, it was hard to follow the rule because of the limitations in finding relevant images. We used 70% of the dataset for training, 20% for validation, and 10% for testing.

5.8.6 Network Configurations

Experiments were conducted on Google Collaboratory professional using the TensorFlow version 1.8 for training, TensorFlow version 1.13.1 for testing, and Keras version 2.1.0 to meet the Mask-

[6] https://www.robots.ox.ac.uk/~vgg/software/via/via_demo.html

RCNN requirements. We used Python for developing training and testing code, and the code used is on this Github repository[7] . One GPU was used while training /testing with two images per GPU. We followed the same default configurations of the Mask-RCNN, so the backbone network is a residual learning framework (resnet101). The learning momentum is 0.9, and the learning rate is 0.001. The weight decay is 0.0001. The batch size is two images per iteration. Images were resized to fit the shape to the dimensions of [1024 1024 3]. Images with four channels were converted to three channels to avoid discrepancies in results. We performed training in 30 epochs with 100 steps per epoch. The average step time is 120 seconds, and the average training loss and validation loss are 0.7905, 0.7034 respectively.

5.8.7 Training Steps

To ease the training process, we transferred learning weights from a pertained Mask R-CNN with MS-COCO [8] dataset. COCO contains datasets used in object detection, pattern detection, and captioning tasks with more than 1.5 million object instances. However, we replaced the last layer of the pertained network with MaxPooling and projection layers. The MaxPooling layer is to reduces the output dimension and projection layer to perform object annotation. The training was performed in three accumulative steps, as discussed below.

5.8.8 Train the MASK-RCNN to Detect Extreme Damage

As a pilot test, we tried first to train the Mask-RCNN network to detect one degree of damage, extreme damage, for the ceiling component. This pilot test ensures that the network can allocate the damaged mask successfully along with the surrounding bounding box, as shown in Figure 5.6. The starting point of the training was transferring the training model ".h5 file" of the COCO dataset. Ninety-two images were used in training, 18 for validation, and six images in testing. We followed the default confidence levels of 0.9 the original Mask-RCNN code.

Results: the network successfully detected extreme damages and located the bounding box and binary mask, as shown in Figure 5.6

[7] https://github.com/samaa/MaskRCNN_Realestate
[8] https://cocodataset.org/#home

5.8.9 Train the Mask-RCNN to Detect the Four Levels of Damages

After the pilot test was performed successfully, we trained the network to detect the four levels of ceiling damage: Extreme, Moderate, Minor and None. A total of 594 degrees of damage were used where an image could contain more than one degree of damage. Also, we tested the network with images with no damage to see how well the network can differentiate between good and damaged ceiling.

Figure 5.6: Extreme Ceiling Damage using Mask-RCNN Showing Instance Segmentation.

5.8.10 Train the MASK-RCNN to Detect Damage Along with Other Objects

This step is important because real estate images are usually occupied with furniture and appliances. These objects help determine what section the image represents. So, the network should be able to differentiate between damages and other objects. For example, an image containing a bathtub is an image of a bathroom section.

We avoided to train the same network to detect other objects such as appliances, plants, light fixtures, kitchen cabinets etc. This option may result in biased training of the network due to unbalanced training samples. In addition, this option requires a huge number of training samples and high computational resources which weren't available. So, we dedicated another network to detect the degree of damage and use another network to detect other objects. We used a general-purpose pre-trained Mask-RCNN with the COCO dataset for general object detection. We followed the same configurations specified for the network in terms of the minimum confidence level accepted by the network.

The developed code aggregated the detection results from the first network, which is dedicated to detecting damages, and the other general-purpose network, as shown in Figure 5.7 below. However, we had to change classes Ids to avoid confusion. For example: class Id one in the general-purpose network is assigned to detect humans, while class Id one was assigned to extreme damage class in the degree of damage network. So, Extreme damage was given another Id to avoid confusion. The results of this training step are shown in Figure 5.7.

Figure 5.7: Detecting Damage with Other Objects in the Image

5.9 Implementing the Search and Retrieval System

In this section, the process of implementing the search and retrieval system using an MCDM is discussed in detail. The process includes defining the goal, criteria, sub-criteria, and how to apply the MCDM to different search listings (alternatives).

5.9.1 AHP Structure of The Real Estate Searching Problem

Before applying AHP, the problem needs to be formulated in the form of an analytical hierarchy process where the goal is defined at the top level, then criteria used in the MCDM process followed by sub-criteria, and finally, the alternatives available to select from. The entire process can be represented in Figure 5.8. The alternatives are a set of listings retrieved from the property database based on the cutoff values specified by the user in the search query. Moreover, if the user has an absolute preference, the search query will be ordered by it. For example:

*select * from property where price between x and y and zipcode in (j, i,k) and has_basment=true order by price, has_basment desc*

the top ten listings retrieved from the search query above are the AHP alternatives. The query will return the properties that have basements, match the price, and are located within the three specified zip codes.

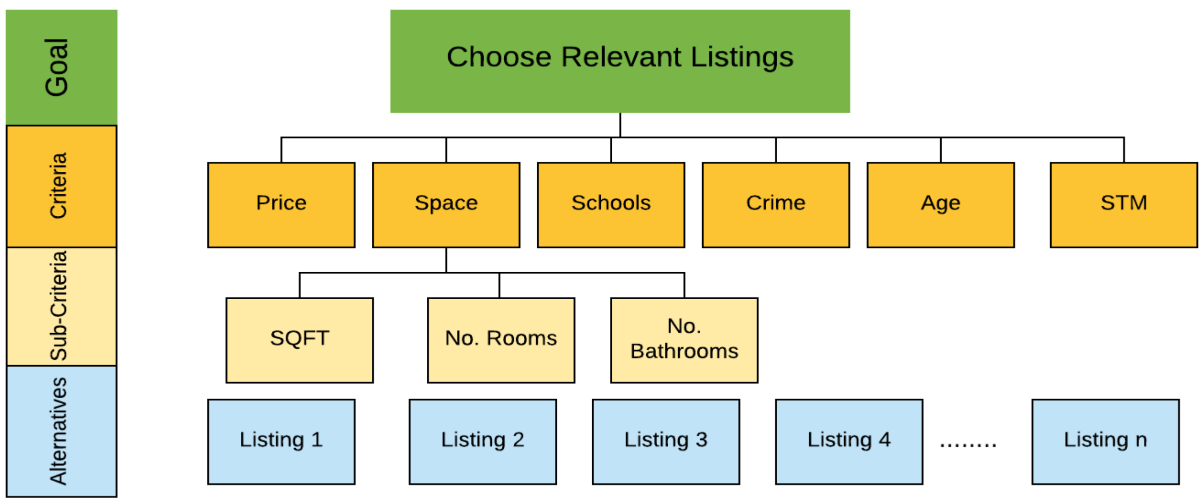

Figure 5.8: Analytical Hierarchy Process for the Real Estate MCDM Problem.

5.9.2 Pair-wise Comparison

The next step is to ask the user to perform pair-wise comparison or preference elicitation where each pair is given four options of preferences of *weak importance, equal importance, stronger importance, and Absolute importance.*

Then, the weight comparison matrix is calculated, and the priority vector is generated by normalizing the eigenvector of the largest eigenvalue. The elements in this eigenvector are the weights of the criteria. The weights are saved in the *User preferences* table of the real-estate website database. The python source code used for the AHP process can be found at this GitHub[9] repository.

5.9.3 Incomplete Pair-Wise Comparison

the pair-wise comparison process might time-consuming task. A combination of paired comparisons C_r^n might be overwhelming when the number of criteria increases. The users may want to avoid some of the comparisons to reduce effort (Bernroider, Maier, & Stix, 2010). Thus, inferring the values of the missing comparisons in the form of simple suggestions and not impositions would help the user during the elicitation process. The Eigenvalue Method (EM), or the Saaty's method (Saaty & Vargas, 1984), maintains desirable rank preservation properties where the principal eigenvector directly captures the rank inherent in the inconsistent data (Harker, 1987). Let $\mathbb{A}$ is an incomplete matrix, and let $\boldsymbol{x} = (x_1, x_2, \ldots, x_k) \in R^k$, be a vector of missing comparisons expressed as criteria of $x_1, x_2, \ldots, x_k$. At this point, the goal is to minimize the inconsistency of the matrix $\mathbb{A}$. So, the EM could be represented as, $\mathbb{A}_x = \lambda_{max}\, x$, where $\lambda_{max}$is the principal eigenvalue of $\mathbb{A}$. For example: if the user sets that the price is extremely important than the space, and the price is extremely important than the location. So, the space might be suggested to be equally important to the location.

5.9.4 Choosing the Best Listings

In this step, the produced weight matrix is used to generate the weight of each criterion in pair comparison between all possible two-listings combinations. The final step is to get their weighted arithmetic sum to yield the rank vector. finally, listings will be ranked in descending order based on their weighted arithmetic sum or rank vector.

5.10 Evaluation

Dunlap, Dotson, and Chambers (1988) conducted the first measure of service quality in real estate brokerage. According to Grönroos (1984), service quality is measured by technical quality and

[9] https://github.com/PhilipGriffith/AHPy

functional quality (Santos, 2003). So, the evaluation process will be divided into two sections. The first is the technical evaluation of the Mask-RCNN degree of damage detector network in terms of the accuracy, training/validation loss, means average precision (mAP). This technical evaluation will show how images details will enhance the *tangibility* of real estate images. The second section is a functional quality evaluation of the search and retrieval system in terms of *flexibility, search results relevancy*, and *efficiency*.

5.10.1 The Evaluation of the Mask-RCNN Degree of Damage Detector

The evaluation of the Mask-RCNN will be in the form of technical testing in terms of accuracy, mean average precision, loss, and sensitivity. Moreover, the *Degree of Damage* detection is a visualized localization of damage which consider an enhancement to image details or the "*tangibles*" of service quality. The degree of damage adds important semantic details to property images, as shown in Figure 5.7, 5.9. The results of the training and testing are presented in table 5.1. These results are calculated at a minimum confidence level of 85 %, which means that the network will detect a degree of damage only if it is at least confident by 85% that the detected damage degree is true. The best Validation accuracy obtained was 93.2% at epoch 24 out of 30 epochs. The network achieved an average testing accuracy of 93 % in detecting damage in the ceiling with an average confidence level of 0.931. Samples of detection results are in Figure 5.9.

To analyze the accuracy of each of the four classes, we plotted the testing accuracy of each class through the 30 epochs as in Figure 5.10. As the number of epochs increases, the average accuracy increases, and the Mask-RCNN confidence in detection increases too as in Figure 5.11. The highest test accuracy was at Epoch 26, where the accuracy started slightly decreasing afterward. The steady increase in the accuracy and the confidence score indicate healthy training and testing processes of the network.

We can notice that the moderate damage class is the most unstable, and it has the least accuracy because some images could fit in both the extreme and the moderate damage classes. On the lower level of the moderate class, some images could fit in the moderate and the minor damage classes.

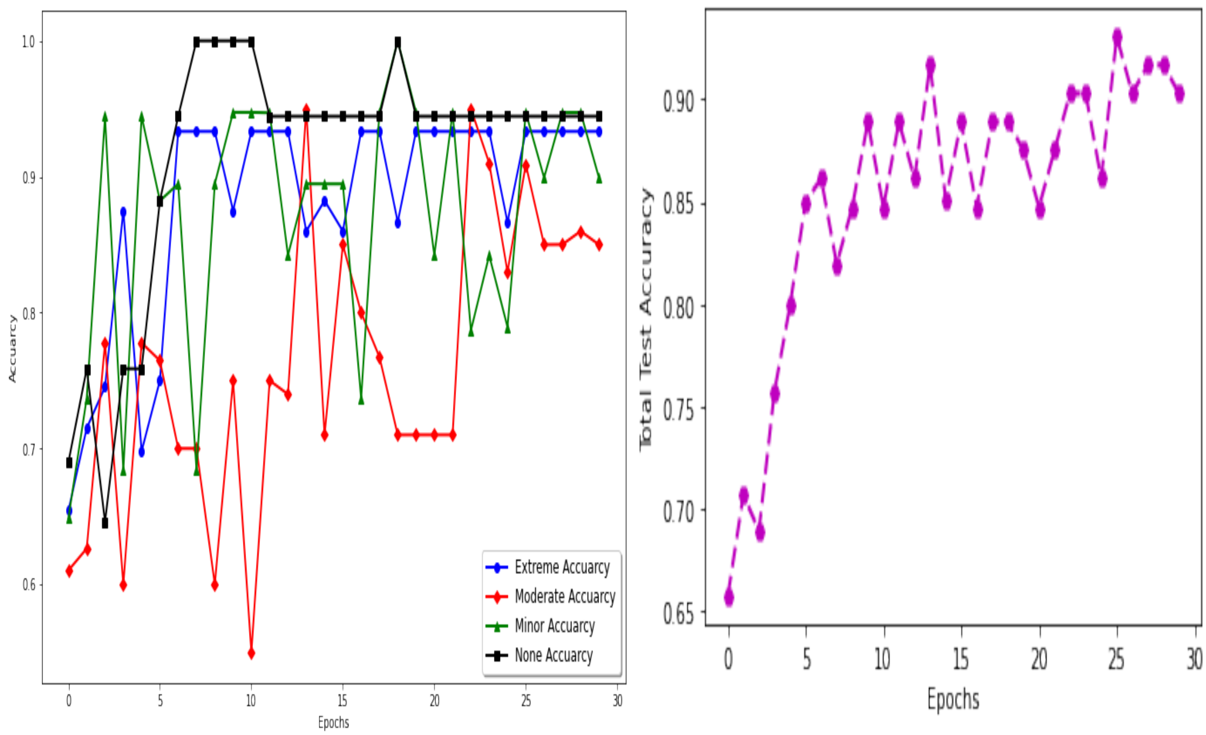

Figure 5.9: Detection Results for Images with Moderate Degree of Damage.

Figure 5.10: The Accuracy of The Four Classes and The Overall Test Accuracy.

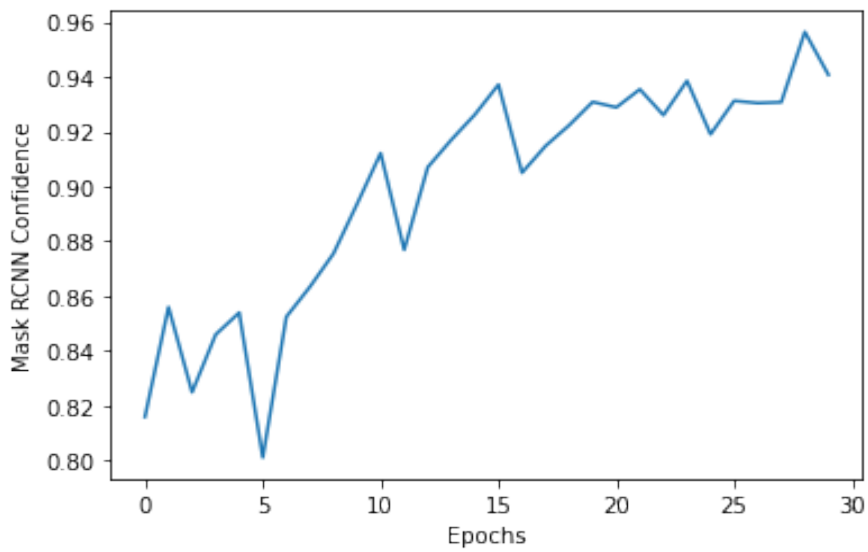

Figure 5.11: Average Confidence of Mask-RCNN in Detecting Test Images

Table 5.1: Validation and Testing Accuracy of the Mask RCNN at 85% Minimum Confidence Level.

	No of training cases	**No of Validation cases**	**No of test cases**	**Validation Accuracy**	**Testing Accuracy**	**Average Confidence validation**
Overall images	594	120	72	0.9325	0.9305	0.9312
Extreme(instances)	171	30	15	0.98187	0.9333	0.9702
Moderate (instances)	200	52	20	0.9722	0.9099	0.8994
Minor (instances)	286	68	19	0.8289	0.9473	0.9311
None (instances)	130	30	18	0.9798	0.9444	0.9299

5.10.2 The Confusion Matrix (Testing)

To get an in-depth analysis of how each class performed. We built the confusion matrix to analyze the precision and Recall of each of the four classes, as in table 5.2.

Table 5.2: the confusion Matrix of predicted classes.

	True Class				
Predicted class		Extreme	Moderate	Minor	None
	Extreme	14	1	0	0
	Moderate	2	18	0	0
	Minor	0	0	18	1
	None	0	0	1	17

1- **Precision:** It tells what fraction of predictions of a positive class were, in fact, positive. To calculate precision, we use the following formula: ***TP/(TP+FP)***. For Extreme Class 0.9333, Moderate 0.9000, Minor 0.947, None 0.9444

2- **Recall:** also known as Sensitivity, *Probability of Detection*. It tells what fraction correctly predicted positive samples is to all positive samples. To calculate Recall, use the following formula: ***TP/(TP+FN).*** For Extreme 0.875, Moderate 0.947, Minor 0.947, and None 0.9444.

5.10.3 Other Performance Measures

Usually, the deep learning network training loss and validation loss are important measures to be considered while training the network. The loss is the cost function we are trying to minimize. At the end of each training epoch, the network evaluates the loss in training and validation. As long as the loss is decreasing, that means that the network accuracy is increasing with each epoch (Jancsary et al., 2012). A rule of thumb is monitoring the validation loss to be slightly less than the training loss. Suppose the training loss is much higher than the validation loss, that's means that the network is overfitting. During the 30 epochs of training, training and validation loss were recorded and plotted as shown in Figure 5.12. We can see that both the training and the test validation start high and they are decreasing gradually. Also, we can notice the slight difference between the validation and training loss in most epochs which indicates that the network didn't go through overfitting during the 30 epochs. The least validation loss was achieved at epoch 26.

The mAP (Mean Average Precision) is a popular metric in measuring the accuracy of object detecting networks such as Faster R-CNN, SSD, etc. The average mean precision computes the average precision value for the recall values over 0 to 1 (Henderson & Ferrari, 2016). For calculating mAP, we have to identify the IoU (Intersection over union). IoU is the ratio of the area of intersection and the area of the union of the ground truth box (the actual bounding box whose coordinates are given in the training set) and predicted bounding boxes. If the IoU is > 0.5, it is considered a True Positive; else, it is considered a false positive.

We calculated the mAP at the end of each training epoch at IoU=0.5, and the maximum mAP achieved was 0.6120. The reason behind this low mAP is that we only provided the mask representing the damage (the polygon) while annotating training images. The highly unstructured nature of the damage made the bounding box relative to the size of the damage. However, the bounding box surrounding the damage wasn't initially specified while training the network.

$$mAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

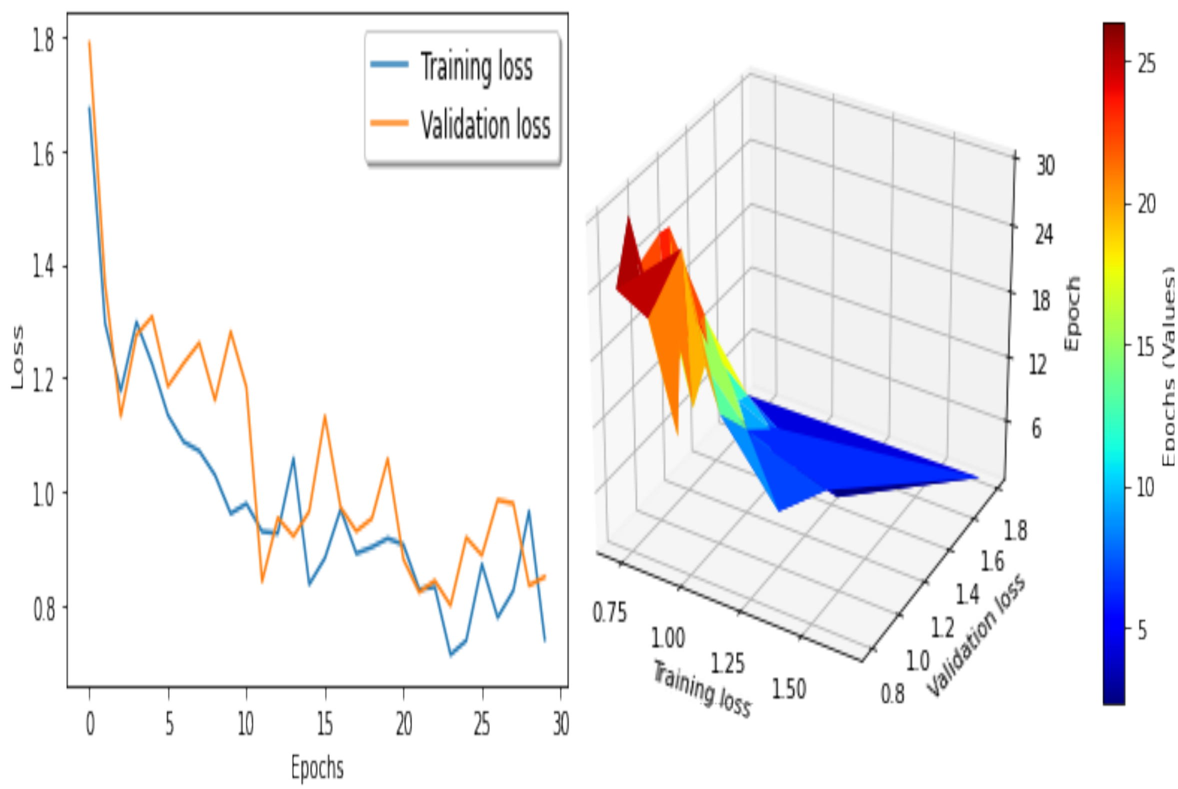

Figure 5.12: Validation Loss Vs. Training Loss.

5.11 Evaluating the Search and Retrieval System

The evaluation of the system will be a functional evaluation in terms of the usage, flexibility, search relevancy, and efficiency of the system. Since the proposed system is a prototype, aesthetic or intangible dimensions of service quality such as linkage, appearance, and structure are not addressed in this research (Gronroos, 1988). So, we focus on the active or tangibles dimensions of service quality, such as information representation, relevancy, efficiency, and flexibility (Xu et al., 2013). The evaluation of the system will be performed using illustrative scenarios to show the use of the system (does the artifact achieve its goal?). Then, we discuss how the system enhances the relevancy and efficiency of the search process.

5.11.1 Illustrative Scenario

Real estate websites in the US usually share a central property database, MLS. The listings in the MCDM filtering website [10] are pulled from this MLS database. These listings are also displayed on popular real estate websites. Two senior couples visited the MCDM filtering website to search for a house or a townhouse. Their priorities are lower crime rate, well-maintained house with low price. Also, newer properties are preferred because the degree of damage is correlated with property age. Their children are grown, so neither schools nor the area of the house are important criteria. This is the first time the couple is using this real estate website to search for properties. According to algorithm 2, the couple used the system as follows:

1- At first, the couple decided the price range, the city of interest, age, and the number of rooms. The system retrieved six relevant results, as shown in Figure 5.13.

2- Then, while the system recommended three criteria to prioritize as in Figure 5.14, the couple performed preference elicitation for six criteria of *price, degree of damage, age, schools, crime, and SQFT*. The couple actively completed all the pair-wise comparisons for the six criteria (15 pair-wise comparisons).

---

[10] https://sites.google.com/view/mcdmrealestate/home

3- Then, user information is saved in the *Users* entity, and the priority vector (criteria weights) of the user is calculated and stored in the *User Preferences* entity to be used in future research.
4- If the degree of damage of any property is not calculated, the property is sent to the degree of damage estimation module, and the degree of damage scores are updated in the *Property Criteria* entity. However, all retrieved properties were previously damage estimated.
5- Then, the system performed a pair-wise comparison between each pair of listings on each criterion, as shown in the Jupitar notebook [11]. The AHP code is written in Python and the system applies AHP with precision of decimal points using Saaty as a random index.

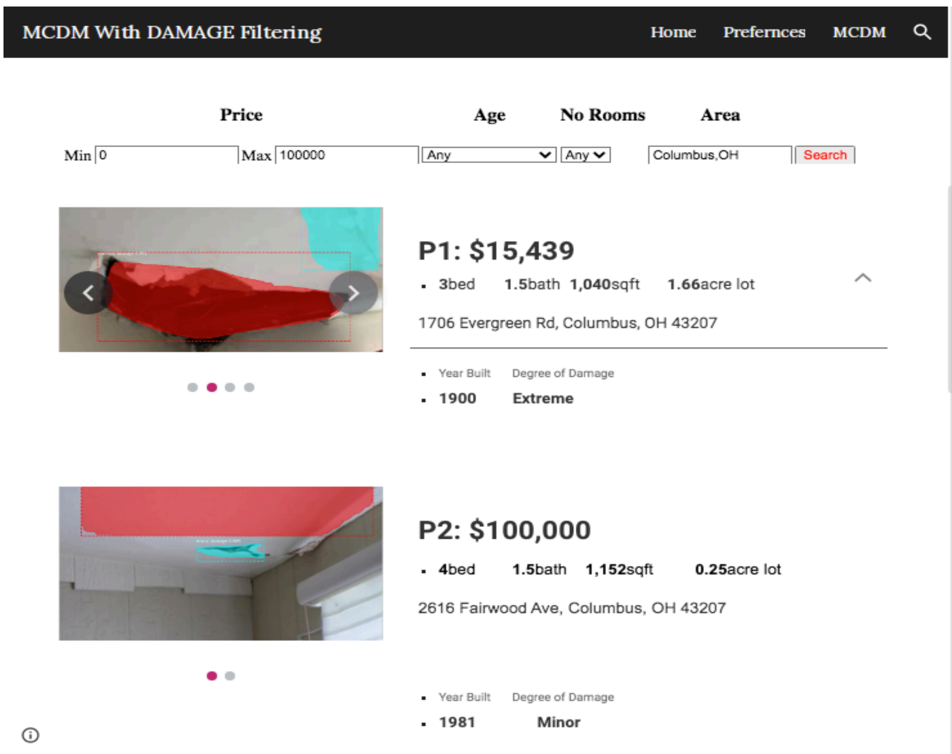

Figure 5.13: Search Results Defining the Price Range, Area, and Age.

6- For each property, the weighted arithmetic sum of criteria is calculated. Then, the listings are ranked in descending order. The ordered rank for the properties that matches the user preferences along with their target weights are: ['p3': 0.205, 'p6': 0.202, 'p2': 0.201, 'p5': 0.165, 'p1': 0.131, 'p4': 0.095].

[11] Colab. Jupiter book

7- A report in JSON Format is created to describe the local weights (*overall local score*) for each criterion and the consistency ratio. We can see that the highest accumulative weight was given to degree of damage followed by crime, year built and price.

8- {"weights": {"local": {"damage": 0.386,"crime": 0.256,"year-built": 0.205,"price": 0.086, "sqft": 0.047, "schools": 0.02} } ,"consistency_ratio": 0.262,"random_index": "Saaty"}

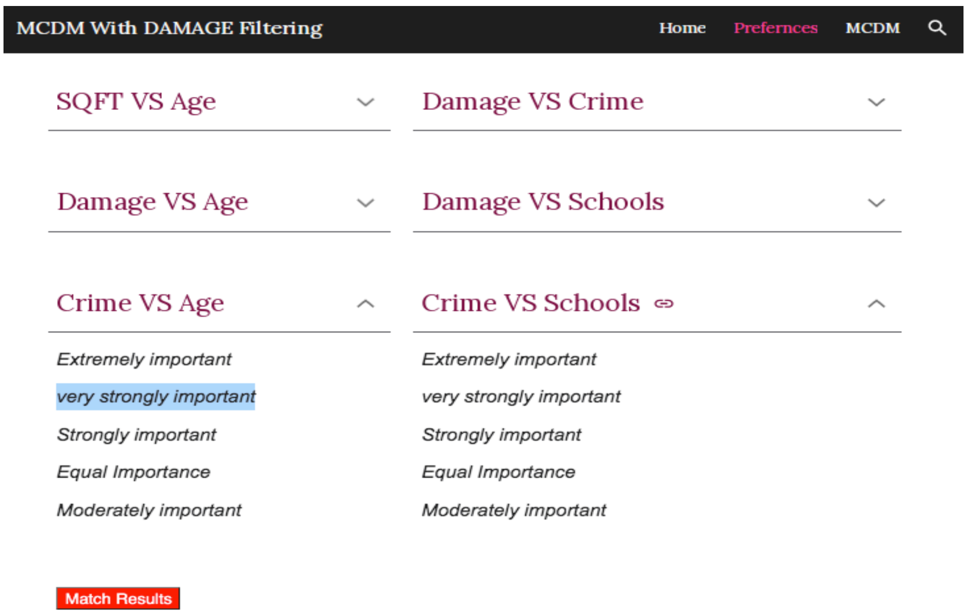

Figure 5.14: Preferences Elicitation for Real Estate MCDM.

9- The displayed listings are reordered based on their matching score (target weights) with specified criteria preferences. Based on the AHP ranking, the listing p3 matches the user preferences the most.

To measure *efficiency*, the couple visited a popular real estate website and entered the same parameters (price range, no. of rooms, age, and location). The website only allows ranking based on price or the date that the listing was added, or lot size. The listing p3 appeared on the second results page after 16 listings. So, the proposed system enhanced the *efficiency* of the search process by finding the desired listing 16 listings earlier.

To measure search results' *relevancy,* we compared the ranked listings with the AHP *overall local score*. The highest criteria *overall local score* was *damage*: 0.386 followed by *crime*: 0.256 and *year-built*: 0.205, i.e., p3 and p6 have "None" *damage* degree; both have the same *crime* rate; however, p3 was given a higher rank because p3 is newer (*year-built*) than p6. So, the system ranked listings are *relevant* to user preferences.

Compared to the other real estate websites, the couple was able to use elucidate six criteria of their preferences, where the regular real estate websites allowed only two criteria of their preferences (price, SQFT). So, the system offered a *flexible* search process where the user could elucidate three times the filtering criteria offered by regular websites. Thus, the system not only allowed *flexible* search experience but also ranked listings *relevant* listings according to user interests towards an *efficient* search process.

5.12 Conclusion

Researchers confirmed the great impact of semantic filters and search tools on enhancing websites' service quality (Yuan et al., 2013) (Ullah & Sepasgozar, 2019). The proposed system introduces the Degree of Damage to the real estate search process. The system also formulates the search process itself into an MCDM problem. The system used a trained Mask-RCNN to detect different degrees of damage in the ceiling. The ceiling is a key component in any real estate section, and it is always assigned a high importance weight because of its high maintenance cost. Due to the lack of a representative degree of damage dataset, the training and testing images used to train the Mask-RCNN network were retrieved from the web search and real-estate websites. The trained network achieved 93% testing accuracy detecting ceiling damage. Additional network was applied in conjunction with the damage detection network to identify the section represented in the image by detecting objects and appliances. i.e., an image with a fridge represents kitchen.

5.12.1 Contributions to Theory

The theorizing process in Design Science Research is to prescribe 'effective development practices' (methods) and 'a type of system solution' (instantiation) for 'a particular class of user requirements' (models) (Markus et al. 2002, p. 180). Such prescriptive theories must be evaluated with respect to the utility provided for the class of problems addressed.". So, knowledge generated as outputs of the design research are both implicit and explicit.

This research contributes to Information Systems research in many ways. The research highlights the significance of including semantic features such as the degree of damage in real estate systems. The knowledge generated from providing semantic feature enhanced the decision process explicitly by adding comprehensive depiction of a property characteristics. The research also highlights the impact of using the new technologies of deep learning and MCDM to enhance real-estate websites' "tangible" dimensions. The "tangible" dimension of service quality has long been ignored despite its significant impact on enhancing service quality. The research adds new search tools and filters to real estate websites that interactively help users fulfill their preferences. The research also shed light on enhancing the decisions support process in key financial services such as real estate search.

As we implemented and evaluated the real estate MCDM, it is important to connect the characteristics of the system in the shadow of IIS characteristics and reflect on how IIS challenges were addressed by the system as discussed in section 1.4, we will discuss each system characteristic and challenge in detail:

Table 5.3: IIS Characteristics and Real Estate MCDM

Characteristic	**How It Is Addressed by The Real estate MCDM**
Interactivity	The proposed real estate system is not only a search system but also a decision-support system. The system asks users to specify their criteria of preferences. The system is flexible and provides multiple criteria for user to elucidate. *Flexibility* always coincides with *Interactivity* (Yadav & Varadarajan, 2005). So, supporting websites' *flexibility* is one method to promote interactivity (Jiang et al., 2010). The system also ranks results based on each listing's weight share of different criteria.
Communication Skills	The system interacts with both the realtor and the user. The system detects damage in images uploaded by the realtor. The system also interacts with the user through performing preference elicitation and re-ranking search results to match user preferences.

Adaptation	The system adapts with old listings. Old listings are not assigned damage degree. However, the system can detect damage in old listings and update their damage score in the database. When the degree of damage is specified as a criteria preference, the system will detect damage in listings before they are displayed to the users.
Predictive Capabilities	The system is designed to predict the degree of damage detected within an image. The system also tries to predict what section is represented in the image, which means that if an image contains a sink and bathtub, the picture most probably represents the bathroom. So, the system predicts the section shown the image from its detected components.

Table 5.4: IIS Challenges and Real Estate MCDM

Challenge	**How It Is Addressed by The Real Estate MCDM**
Responsiveness	Responsiveness is usually associated with time, or how long the system takes to respond to a user request. However, responsiveness can be viewed from two angles. The first is the pace of response, and the second it the quality of response or how the system match user expectations (Kaufman, 1991). In other words, how the proposed real estate system responds to user's search requests as expected by the user. The system used MCDM as a channel where users could submit preferred listings' criteria and the system ranks the listings to best match their preferred criteria.
Interactivity Level	The system presented two levels of interactivity. The first is on the kernel level, where the system supports the decision process of selection between different listings. The preference elicitation feature allows user to prioritize their criteria of preferences so the system could rank properties accordingly. The system also interacts on the apparent level where the system visualize

	damage in real estate listings images when the user selects degree of damage as criterion of preferences.
Efficiency	Popular real estate websites offer limited number of search filters and limited sort options. The proposed website allows listings ranking based on variable number of search filters where filters are given unequal importance weights. So, listings of interest are displayed in the first page of results while these listings are displayed dispersedly in late search results pages in most popular real estate websites.
Tangibility	The degree of damage is a adding a visual dimension to the real estate service quality where damages are identified and quantified. The degree of damage provides significant information that help making comprehensive real estate decisions.

5.12.2 Limitations and Future Research

The evaluation of the trained Mask-RCNN network revealed that the highest accuracy classes are the extreme damage and the no damage class, while the moderate damage class achieved the least accuracy. The reason why moderate damage scored the least accuracy is the broad spectrum of damage degrees that could fit the moderate damage class. So, some damages could fit both moderate/minor class or moderate/extreme. If we looked in-depth, we could find that big dark moldy spots are usually detected as extreme damage rather than moderate. Therefore, real estate experts' contributions are needed to define the damage spectrum for each degree of damage. On the other hand, the none and extreme damage classes scored the highest accuracy because they represent the beginning and end of the damage spectrum.

The new degree of damage is adding a new quality measure to the "tangibles" dimension of service quality. The research used AHP, a popular intuitive MCDM process, to assist users in matching their criteria of preferences. The user assigns weights to the criteria of preference in pair-wise comparison with other criteria. Then, the priority vector of the user preferences is calculated and stores in the *User Preferences* database entity. The MCDM includes more inclusive criteria not found in the regular search process, which promotes flexibility and efficiency. The search results are ordered according to their weighted score in descending order. However, if the number of retrieved listings is large, MCDM might take a long time to create a ranked list that matches user preferences. Therefore, the initial ordering of properties is based on the absolute criteria of preferences helped accelerated the MCDM process. Overall, matching search results with user preference enhances the search process efficiency, information relevancy that in turn enhances the service quality, trust, and user satisfaction.

While the trained network achieved 93% accuracy for detecting damages, the determination of what is considered minor, moderate, and extreme is very challenging and subjective. There are some cases where damage could fit into two damage degrees. For example, a big moldy spot could fit both moderate and extreme damage degrees. To address these challenges, the involvement of real-estate inspectors is required to specify the criteria range for each degree of damage.

Due to the hardship of getting a relevant dataset for damages in real estate, the training, testing, and validation samples were limited to detect the degree of damage in the ceiling. Therefore, more training samples to detect damage in different components and sections are required to allow a comprehensive evaluation of a property.

These ceiling objects such as air vents, smoke detectors, speakers, and light fixtures need to be considered as additional classes in the training process, which will require strong GPUs to train multiple classes in the ceilings alone. Therefore, for comprehensive and precise damage detection, all fixtures in real estate images should be included in the training process. This step requires a huge number of training samples for each component in different degrees of damage.

Future work will include developing an inclusive degree of damage real estate dataset. The dataset should include all real estate components and sections in different degrees of damage. Developing this dataset should involve real estate experts to carefully define the criteria for annotating training images to be assigned the correct degree of damage. It is important to include the source of damage, like the water leak or the paint cracks, in quantifying the damage. Even for the same degree of damage, usually, the source of damage could have a great impact on the property. For example: moderate damage caused by paint cracks could be cheaper to fix than moderate damage caused by moldy areas. However, this step requires including real estate inspectors and experts to identify the source of damages appropriately.

Chapter 6 Conclusion

6.1 Future Work

For the real estate search system, Future work includes developing an inclusive degree of damage real estate dataset. The dataset should include all real estate components and sections in different states of damage. Developing this dataset should involve real estate experts to carefully define the criteria for annotating training images to be assigned the correct degree of damage.

The system focused on detecting damage in the ceiling. In the next step, the training and testing of the network include different components and sections. This step requires many training samples from each component and section in different degrees of damage. In this case, the network needs high computational resources beyond Google Collaboratory.

It is important to include the source of damage, such as water leaks or paint cracks, in quantifying the damage. Even for the same degree of damage, usually, the source of damage could impact the cost of maintenance. For example, moderate damage caused by paint cracks could be cheaper to fix than moderate damage caused by dark moldy areas. However, this step requires including real estate inspectors and experts to identify the source of damages appropriately.

For the SAFEMYRIDES system, Future work includes implementing the system on other platforms such as IOS, Raspberry Pi. The system should be tested on different platforms to compare and contrast its performance and stability. In addition, the size of the violation's dataset is very limited. Increasing the training and validation samples will enhance the accuracy and decrease the latency of predictions.

Encryption of the violation incidents was performed using a simple XOR function. More sophisticated encryption methods using a 128-Bit key are preferred. Since the system achieved lower accuracy and decreased sensitivity in detecting violations during the night, the violation detection network needs to be trained with more samples during the night where poor light obstructs well-defined detection of violations.

Benchmarking for the SAFEMYRIDES system is required because the detection accuracy depends on the specification of the phone cameras. Benchmarking will help to determine the minimum required specifications in order to obtain reasonable detection accuracy. Further research on the optimization of cameras usage is required in terms of decreasing camera battery consumption, assuming that a ridesharing driver will be using the system for hours. Moreover, the system should be deployed in a real setting to address other intangibles service quality measures such as reliability, understanding, and courtesy.

6.2 Addressing IIS Characteristics

To attain a big picture about what are the IIS characteristics that were used in designing the proposed systems, as discussed in section 1.4, we will discuss each IIS characteristic in detail:

Table 6.1: IIS Characteristics Addressed by The Research

Characteristic	**How It Is Addressed**
Interactivity	the proposed systems are both interactive. They provide decision-support/making for the users. The SAFEMYRIDES interacts with users and ask their permissions to monitor ridesharing session to maintain user privacy. The system also voices out a warning in case of detected violation, so everyone is aware that a violation is reported. The real estate MCDM system provide decision-support for real estate websites' users. The system provides a flexible pool of filtering criteria for user to prioritize. Then, the system re-rank listings to match user criteria of preferences.
Communication Skills	The ridesharing system interacts with system users in a verbal and a written format. The real estate MCDM interacts with both the realtor and the user. The real estate system interacts with the user by performing preference elicitation and re-ranking search results to match user preferences. While the communication skills used in both systems are limited to certain functions, they are considered appropriate according to system requirements to avoid overwhelming users (Augello, Gentile, & Dignum, 2017).

Adaptation	Both systems can adapt certain changes in the environment. The Ridesharing system can adapt the light change from day to night during detection, and it can be used with rear or front cameras. The real estate MCDM can adapt with old listings. The system can detect damage in old listings and update their damage score in the database. When a damage is specified as a criteria preference, the system detect damage in listings before they are displayed to the users.
Predictive Capabilities	Both systems use deep learning models that are trained to predict or identify certain object or events. In the ridesharing system, the deep learning model try to predict which person in the ridesharing session is a driver and which is the passenger. It also tries to predict if a certain action is considered a violation or not. The real estate system is designed to predict the degree of damage of certain damages within an image. The system also tries to predict what section is represented in the image.

6.3 Addressing IIS Challenges

Each of the developed systems is trying to address the IIS challenges as discussed in section 1.4. So, we are trying to summarize how these challenges were addressed by the two systems as discussed below:

Table 6.2: IIS Challenges Addressed by The Research

Challenge	**How It Is Addressed**
Responsiveness	Responsiveness is not only a quantitative measure, or how long the system takes to respond to a user request. Responsiveness can be viewed from two

	angles. The first is the pace of response, and the second it the quality of response or how the system match user expectations (Kaufman, 1991). In the ridesharing system, the average system *Latency* was used as a measure of the system responsiveness. The average response time to detect violations is 620 ms which is the least among the current systems. The real estate system responsiveness was measured by how the system responded to user's search requests as expected by the user. The system used MCDM as a channel where users could submit desired listings criteria and the system ranks the listings to best match their desired criteria.
Privacy	Privacy was a challenge in building the ridesharing system. The main question that needs an answer is how to maintain confidentiality while monitoring ridesharing. Transparency and violation encryption were used to solve the privacy issues. The system speaks out "violation detected", so both the driver and the passenger are aware that the ridesharing session is monitored. Also, the authorization of users is required to monitor ridesharing sessions. Violation incidences are maintained confidential by encrypting violation incidences before they are reported to ridesharing agents.
Interactivity Level	The proposed systems interact with users on the kernel level within the core system activities. In the ridesharing system, a verbal warning is displayed when a violation is detected, which adds a core feature in the monitoring systems. The system is transparent about recoding an incidence where users are aware of recording. In the real estate system, the preference elicitation feature allowed users to prioritize their criteria of preferences so the system could rank properties accordingly.
Efficiency	It is expected from systems that run deep learning models to be resources exhaustive. So, it is challenging to build a system that runs a deep learning

	model and is not resources extensive. The ridesharing system used only 29.09 MB of the phone's internal storage and a Memory (RAM) usage was just 104 MB despite running real-time inference, which means that the system is consuming less than 1% of the smartphone RAM of 8 GB. The proposed real estate MCDM system allows users to use variable number of search filters where filters are given unequal importance weight. So, properties of interest are listed in the first page of results, while they are listed dispersedly in late search pages in popular real estate websites
Tangibility	Tangibility is the visual characteristic of the service provided. The ridesharing system adds another tangibility dimension to the ridesharing services by adding a camera streams that run object detection continuously during the monitoring process. Also, the degree of damage is a new visual dimension to the real estate service quality where damage is identified and quantified

6.4 Contribution to Theory and Society

The SAFEMYRIDES system is trying to solve the safety issues related to ridesharing, and it has several implications to practice. Firstly, the system allows local ridesharing monitoring on the driver's smartphone, saving the cost of an additional device for monitoring. Having the monitoring system running on the same device used in the ridesharing will ensure efficiency and privacy. Secondly, the system uses no network communication while detecting violations and limits the network communication to reporting the violation to the ridesharing agent. Minimizing network communication ensures privacy, security, and efficiency. So, the system could be used where poor network connections exist. Thirdly, encryption of ridesharing incidences ensures privacy and security. So, the passenger/driver data are only shared directly with the ridesharing agent in case of detected violation.

The implication of the real estate MCDM system to practice include enhancing the real estate websites users experience through enhancing flexibility, efficiency, and information relevancy. Moreover, introducing new semantic features such as the degree of damage enhances the information quality provided to users to make informed decisions. The system also uses MCDM to support users' decisions by matching their criteria of preference, enhancing service quality and customer satisfaction.

Both Artifacts contribute to Information Systems research in many ways. The SAFEMYRIDES applies rigorous technical methods such as decentralized control, where an IoT (smartphone) can independently detect violations and make decisions. SAFEMYRIDES is one of the early attempts to apply different optimization methods to deep learning networks to run efficiently on an IoT. The research sheds light on the importance of limiting cloud/edge computation and limiting network communications while enhancing the decisions making process and analytics of the IoTs.

The Real estate MCDM system highlights the significance of including semantic features such as the degree of damage in real estate systems. The research also highlights the impact of using the new technologies of deep learning and MCDM to enhance real-estate websites' "tangible" dimensions. The "tangible" dimension of service quality has long been ignored despite its significant impact on enhancing service quality. The real estate system adds new search tools and filters to real estate websites that interactively help users fulfill their preferences.

6.5 Summary

Intelligence is a core component in current Intelligent Information systems (IIS). Among the abundance of AI technologies, IoT, Edge Computing, and deep learning got notable research attention because of their promising contribution to many industries. However, studying the potential of employing these technologies to overcome IIS challenges still not addressed.

The dissertation is divided into two IISs. The first is building SAFEMYRIDES, a scene-aware system for monitoring ridesharing. The system uses optimized deep learning models for IoT that can detect violations in ridesharing locally on the driver's smartphone. The system records only violations indicine in an encrypted format. Implementing ridesharing monitoring services faces many technical challenges such as high cost, complexity, and network dependency. In addition, monitoring the entire ridesharing session violates the user's privacy. The system enhances the ridesharing monitoring quality in terms of *responsiveness, privacy, and efficiency*. The system architecture is simple and decreases the cost of connection to cellular internet networks required for monitoring ridesharing sessions. On the social level, the system would ensure privacy while preventing hundreds of crimes and which enhances customer satisfaction and safety without violating privacy.

The second IIS aims at enhancing the service quality of real-estate systems by adding more semantic search tools and filters. We developed a new real estate filter called the degree of damage, which is detected visually from real estate images using *Deep Learning* for computer vision. In addition, we developed an enhanced real estate search system that filters real estate as a multi-criteria decision problem, using the Analytical hierarchal process (AHP). The proposed system enhances the real estate agents' service quality by enhancing flexbilty and the relevancy of the search results and process. Including the degree of damage in filtering real estate enhances the accessibility of information and the user experience.

Reference

Abdel-Basset, M., Mohamed, R., Sallam, K., & Elhoseny, M. (2020). A novel decision-making model for sustainable supply chain finance under uncertainty environment. *Journal of Cleaner Production, 269*, 122324.

Acharya, R., Kagan, A., & Zimmerman, T. (2010). Real estate information technology: Influence of email marketing on real estate agent performance. *Journal of Real Estate Literature, 18*(2), 329-343.

Aghnout, S., & Karimi, G. (2019). Modeling triplet spike timing dependent plasticity using a hybrid TFT-memristor neuromorphic synapse. *Integration, 64*, 184-191.

Aken, J. E. v. (2004). Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. *Journal of management studies, 41*(2), 219-246.

Akerkar, R. (2012). Intelligent systems: perspectives and research challenges. *CSI Communications, 5*.

Al Kilani, M., & Kobziev, V. (2016). An Overview of Research Methodology in Information System (IS). *Open Access Library Journal, 3*(11), 1-9.

Alba, J., Lynch, J., Weitz, B., Janiszewski, C., Lutz, R., Sawyer, A., & Wood, S. (1997). Interactive home shopping: consumer, retailer, and manufacturer incentives to participate in electronic marketplaces. *Journal of marketing, 61*(3), 38-53.

Ali, B. J., Saleh, P. F., Akoi, S., Abdulrahman, A. A., Muhamed, A. S., Noori, H. N., & Anwar, G. (2021). *Impact of Service Quality on the Customer Satisfaction: Case study at Online Meeting Platforms.* Paper presented at the Ali, BJ, Saleh, Akoi, S., Abdulrahman, AA, Muhamed, AS, Noori, HN, Anwar, G.(2021). Impact of Service Quality on the Customer Satisfaction: Case study at Online Meeting Platforms. International journal of Engineering, Business and Management.

Almoqbel, M. Y., & Wohn, D. Y. (2019). Individual and Collaborative Behaviors of Rideshare Drivers in Protecting their Safety. *Proceedings of the ACM on Human-Computer Interaction, 3*(CSCW), 1-21.

Amaral, M., Polo, J., Carrera, D., Seelam, S., & Steinder, M. (2017). *Topology-aware gpu scheduling for learning workloads in cloud environments.* Paper presented at the Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis.

Anderson, D. N. (2016). Wheels in the head: ridesharing as monitored performance. *Surveillance & Society, 14*(2), 240-258.

Armant, V., & Brown, K. N. (2020). Fast optimised ridesharing: Objectives, reformulations and driver flexibility. *Expert Systems with Applications, 141*, 112914.

Arnott, D. (2006). Cognitive biases and decision support systems development: a design science approach. *Information Systems Journal, 16*(1), 55-78.

Aronson, J. E., Liang, T.-P., & Turban, E. (2005). *Decision support systems and intelligent systems* (Vol. 4): Pearson Prentice-Hall New York.

Arteaga-Sánchez, R., Belda-Ruiz, M., Ros-Galvez, A., & Rosa-Garcia, A. (2020). Why continue sharing: Determinants of behavior in ridesharing services. *International Journal of Market Research, 62*(6), 725-742.

Athalye, S. (2013). Recommendation system for news reader.

Augello, A., Gentile, M., & Dignum, F. (2017). *An overview of open-source chatbots social skills.* Paper presented at the International conference on internet science.

Aw, E. C.-X., Basha, N. K., Ng, S. I., & Sambasivan, M. (2019). To grab or not to grab? The role of trust and perceived value in on-demand ridesharing services. *Asia Pacific Journal of Marketing and Logistics*.

Badriyah, T., Azvy, S., Yuwono, W., & Syarif, I. (2018). *Recommendation system for property search using content based filtering method.* Paper presented at the 2018 International conference on information and communications technology (ICOIACT).

Balaž, Z., & Predavec, D. (2018). *Integration of Cognitive Cybernetics into Intelligent Human Systems.* Paper presented at the International Conference on Intelligent Human Systems Integration.

Beck, U., Lash, S., & Wynne, B. (1992). *Risk society: Towards a new modernity* (Vol. 17): sage.

Bernroider, E. W., Maier, K., & Stix, V. (2010). *Incomplete information within relative pairwise comparisons as utilized by the AHP.* Paper presented at the International Conference on Business Information Systems.

Bhardwaj, A., Di, W., Hamid, M. R., Piramuthu, R., & Sundaresan, N. (2018). Extraction of image feature data from images. In: Google Patents.

BILL, H. Passenger Protection Act. Retrieved from https://www.ncleg.gov/Sessions/2019/Bills/House/PDF/H391v7.pdf

Bin, J., Gardiner, B., Li, E., & Liu, Z. (2019). Peer-Dependence Valuation Model for Real Estate Appraisal. *Data-Enabled Discovery and Applications, 3*(1), 2.

Blayse, A. M., & Manley, K. (2004). Key influences on construction innovation. *Construction innovation*.

Boley, H., & Chang, E. (2007). *Digital ecosystems: Principles and semantics.* Paper presented at the 2007 Inaugural IEEE-IES Digital EcoSystems and Technologies Conference.

Boos, D., Guenter, H., Grote, G., & Kinder, K. (2013). Controllable accountabilities: the internet of things and its challenges for organisations. *Behaviour & Information Technology, 32*(5), 449-467.

Bowonder, B., & Miyake, T. (1992). A model of corporate innovation management: some recent high tech innovations in Japan. *R&D Management, 22*(4), 319-336.

Brous, P., & Janssen, M. (2015). *Advancing e-Government using the internet of things: a systematic review of benefits.* Paper presented at the International Conference on Electronic Government.

Brunelli, M. (2014). *Introduction to the analytic hierarchy process*: Springer.

Carlsson, S. A. (2006). *Towards an information systems design research framework: A critical realist perspective.* Paper presented at the Proceedings of the First International Conference on Design Science Research in Information Systems and Technology, Claremont, CA.

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature News, 538*(7623), 20.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST), 2*(3), 1-27.

Chatterjee, P., & Armentano, R. L. (2015). *Internet of things for a smart and ubiquitous eHealth system.* Paper presented at the 2015 International Conference on Computational Intelligence and Communication Networks (CICN).

Chaudhry, B., El-Amine, S., & Shakshuki, E. (2018). Passenger safety in ride-sharing services. *Procedia computer science, 130*, 1044-1050.

Chen, G., Zhao, Y., Shen, X., & Zhou, H. (2017). *Effisha: A software framework for enabling effficient preemptive scheduling of gpu.* Paper presented at the Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming.

Chen, J., Hui, E. C., & Wang, Z. (2011). Perceived risk, anticipated regret and post-purchase experience in the real estate market: The case of China. *Housing Studies, 26*(03), 385-402.

Cheng, X., Bao, Y., Zarifis, A., & Mou, J. (2019). A model of customer trust in sharing economy-driven ride-sharing platforms involving psychological contract violation and recovery.

Cheng, X., Su, L., Luo, X., Benitez, J., & Cai, S. (2021). The good, the bad, and the ugly: impact of analytics and artificial intelligence-enabled personal information collection on privacy and participation in ridesharing. *European Journal of Information Systems*, 1-25.

Cheung, S.-C., Venkatesh, M. V., Paruchuri, J. K., Zhao, J., & Nguyen, T. (2009). Protecting and managing privacy information in video surveillance systems. In *Protecting Privacy in Video Surveillance* (pp. 11-33): Springer.

Chimbalkar, A., Patil, S., Jain, M., Kate, G., & Dwivedi, P. (2014). Finding n-Most Demanding Products for Real-Estate Industry. *International Journal of Computer Applications, 93*(12).

Chiu, Y.-C., Tsai, C.-Y., Ruan, M.-D., Shen, G.-Y., & Lee, T.-T. (2020). *Mobilenet-SSDv2: An improved object detection model for embedded systems.* Paper presented at the 2020 International Conference on System Science and Engineering (ICSSE).

Choi, Y. J., & Kim, S.-I. (2017). A Study on the Usability of Mobile Real Estate Service Search Filter and Information Quality-Focused on domestic, US and UK cases. *Journal of the Korea Convergence Society, 8*(9), 249-254.

Ciuna, M., De Ruggiero, M., Manganelli, B., Salvo, F., & Simonotti, M. (2017). *Automated valuation methods in atypical real estate markets using the mono-parametric approach.* Paper presented at the International Conference on Computational Science and Its Applications.

Ciuna, M., Salvo, F., & Simonotti, M. (2017). The multilevel model in the computer-generated appraisal: a case in Palermo. In *Advances in Automated Valuation Modeling* (pp. 225-261): Springer.

Cleven, A., Gubler, P., & Hüner, K. M. (2009). *Design alternatives for the evaluation of design science research artifacts.* Paper presented at the Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology.

Consortium, W. W. W. (2002). The platform for privacy preferences 1.0 (P3P1. 0) specification.

Crabtree, B. F., & Miller, W. L. (1992). *Doing qualitative research.* Paper presented at the Annual North American Primary Care Research Group Meeting, 19th, May, 1989, Quebec, PQ, Canada.

Crowder, J. A., Carbone, J., & Friess, S. (2020). Systems-Level Thinking for Artificial Intelligent Systems. In *Artificial Psychology* (pp. 15-27): Springer.

d'Amato, M. (2018). Supporting property valuation with automatic reconciliation. *Journal of European Real Estate Research*.

Dabholkar, P. A., & Overby, J. W. (2005). Linking process and outcome to service quality and customer satisfaction evaluations. *International journal of service industry management*.

Dabholkar, P. A., Shepherd, C. D., & Thorpe, D. I. (2000). A comprehensive framework for service quality: an investigation of critical conceptual and measurement issues through a longitudinal study. *Journal of retailing, 76*(2), 139-173.

Davenport, T. H., & Kudyba, S. (2016). Designing and developing analytics-based data products. *MIT Sloan Management Review, 58*(1), 83.

Del Giudice, V., De Paola, P., & Forte, F. (2017). Using genetic algorithms for real estate appraisals. *Buildings, 7*(2), 31.

DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: a ten-year update. *Journal of management information systems, 19*(4), 9-30.

Demetis, D. S., & Lee, A. S. (2016). Crafting theory to satisfy the requirements of systems science. *Information and Organization, 26*(4), 116-126.

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing, 7*(3–4), 197-387.

Diffie, W., & Landau, S. (2010). *Privacy on the line: The politics of wiretapping and encryption*: The MIT Press.

Dillahunt, T. R., Kameswaran, V., Li, L., & Rosenblat, T. (2017). *Uncovering the values and constraints of real-time ridesharing for low-resource populations.* Paper presented at the Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.

Dills, A. K., & Mulholland, S. E. (2018). Ride-sharing, fatal crashes, and crime. *Southern Economic Journal, 84*(4), 965-991.

Domingos, P. (2015). *The master algorithm: How the quest for the ultimate learning machine will remake our world*: Basic Books.

Duan, Y., Edwards, J. S., & Dwivedi, Y. K. (2019). Artificial intelligence for decision making in the era of Big Data–evolution, challenges and research agenda. *International Journal of Information Management, 48*, 63-71.

Dunlap, B. J., Dotson, M. J., & Chambers, T. M. (1988). Perceptions of real-estate brokers and buyers: A sales-orientation, customer-orientation approach. *Journal of Business research, 17*(2), 175-187.

Eiband, M., Buschek, D., & Hussmann, H. (2021). *How to support users in understanding intelligent systems? structuring the discussion.* Paper presented at the 26th International Conference on Intelligent User Interfaces.

Eisenmeier, S. (2019). Ride Sharing Platforms in Developing Countries. Effects and Implications for Mexico City. In: New York: Oxford University Press.

El Khaddar, M. A., & Boulmalf, M. (2017). Smartphone: the ultimate IoT and IoE device. *Smartphones from an applied research perspective, 137*.

Elhoseny, M., Metawa, N., Darwish, A., & Hassanien, A. E. (2017). Intelligent information system to ensure quality in higher education institutions, towards an automated e-university. *International Journal of Computational Intelligence Studies, 6*(2-3), 115-149.

Elnagar, S., & Osei-Bryson, K.-M. (2020). *Towards Applying Deep Learning to the Internet of Things: A Model and a Framework.* Paper presented at the European, Mediterranean, and Middle Eastern Conference on Information Systems.

Elnagar, S., & Thomas, M. A Cognitive Framework for Detecting Phishing Websites.

Elnagar, S., & Thomas, M. A. (2019). Real Estate Image-Based Appraisal Using Mask Region Based Convolutional Networks.

Elnagar, S., & Thomas, M. A. (2020). Federated Deep Learning: A Conceptual Model and Applied Framework for Industry 4.0.

Elnagar, S., & Weistroffer, H. R. (2019). *Introducing knowledge graphs to decision support systems design.* Paper presented at the Eurosymposium on systems analysis and design.

Emmert-Streib, F., Yli-Harja, O., & Dehmer, M. (2020). Explainable artificial intelligence and machine learning: A reality rooted perspective. *arXiv preprint arXiv:2001.09464*.

Essays. (2018). Decision support systems. Retrieved from Retrieved from https://www.ukessays.com/essays/it-research/decision-support-systems.php?vref=1

Feeney, M. (2015). Is ridesharing safe? *Cato Institute Policy Analysis*(767).

Figueira, J. R., Greco, S., Roy, B., & Słowiński, R. (2013). An overview of ELECTRE methods and their recent extensions. *Journal of Multi‐Criteria Decision Analysis, 20*(1-2), 61-85.

Frajberg, D., Bernaschina, C., Marone, C., & Fraternali, P. (2019). *Accelerating Deep Learning inference on mobile systems.* Paper presented at the International Conference on AI and Mobile Services.

Garcia, A., & O'Brien, S. (2019). Uber releases safety report revealing 5,981 incidents of sexual assault. Retrieved from https://www.cnn.com/2019/12/05/tech/uber-safety-report

Gargallo Valero, P., Salvador Figueras, M., & Miguel Álvarez, J. Á. (2017). *MCMC Bayesian spatial filtering for hedonic models in real estate markets*. Retrieved from

Gerring, P. E., Shortliffe, E. H., & van Melle, W. (1982). Interviewer/Reasoner Model: An Approach to Improving System Responsiveness in Interactive AI Systems. *AI Magazine, 3*(4), 24-24.

Ghoshal, S., & Moran, P. (1996). Bad for practice: A critique of the transaction cost theory. *Academy of management Review, 21*(1), 13-47.

Goldman, E. (2005). Deregulating relevancy in Internet trademark law. *Emory LJ, 54*, 507.

Golstein, B. (2018). A brief taxonomy of AI. In: Obtenido de Sharper AI: https://www. sharper. ai/taxonomy-ai.

Gregor, S. (2006). The nature of theory in information systems. *MIS quarterly*, 611-642.

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information systems, 8*(5), 312-335.

Gretzel, U. (2011). Intelligent systems in tourism: A social science perspective. *Annals of tourism research, 38*(3), 757-779.

Gronroos, C. (1988). Service quality: The six criteria of good perceived service. *Review of business, 9*(3), 10.

Grönroos, C. (1984). A service quality model and its marketing implications. *European Journal of marketing*.

Guerlain, S., Brown, D. E., & Mastrangelo, C. (2000). *Intelligent decision support systems.* Paper presented at the Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0.

Guevara, L., & Auat Cheein, F. (2020). The Role of 5G Technologies: Challenges in Smart Cities and Intelligent Transportation Systems. *Sustainability, 12*(16), 6469.

Gupta, S., Buriro, A., & Crispo, B. (2019). DriverAuth: A risk-based multi-modal biometric-based driver authentication scheme for ride-sharing platforms. *Computers & Security, 83*, 122-139.

Hallerbach, W. G., & Spronk, J. (2002). The relevance of MCDM for financial decisions. *Journal of Multi-Criteria Decision Analysis, 11*(4-5), 187-195.

Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

Harker, P. T. (1987). Incomplete pairwise comparisons in the analytic hierarchy process. *Mathematical modelling, 9*(11), 837-848.

Hart, D., & Gregor, S. (2010). *Information systems foundations: the role of design science*: ANU E Press.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *Mask r-cnn.* Paper presented at the Proceedings of the IEEE international conference on computer vision.

Henderson, P., & Ferrari, V. (2016). *End-to-end training of object class detectors for mean average precision.* Paper presented at the Asian Conference on Computer Vision.

Hevner, A., & Chatterjee, S. (2010). Design science research in information systems. In *Design research in information systems* (pp. 9-22): Springer.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.

Hizam, S. M., & Ahmed, W. (2020). A conceptual paper on SERVQUAL-framework for assessing quality of Internet of Things (IoT) services. *arXiv preprint arXiv:2001.01840*.

Ho, C.-I., & Lee, Y.-L. (2007). The development of an e-travel service quality scale. *Tourism management, 28*(6), 1434-1449.

Holmström, J., Ketokivi, M., & Hameri, A. P. (2009). Bridging practice and theory: A design science approach. *Decision Sciences, 40*(1), 65-87.

Hong, S. J. (2017). Assessing economic value of reducing perceived risk in the sharing economy: the case of ride-sharing services.

Huang, E. Y., Lin, S.-W., & Fan, Y.-C. (2015). MS-QUAL: Mobile service quality measurement. *Electronic Commerce Research and Applications, 14*(2), 126-142.

Huang, J.-C., Huang, H.-C., & Liu, H.-H. (2019). Research on the Parallelization of Image Quality Analysis Algorithm Based on Deep Learning. *Journal of Visual Communication and Image Representation*, 102709.

Huang, Z., Zhong, Z., Sun, L., & Huo, Q. (2019). *Mask R-CNN with pyramid attention network for scene text detection.* Paper presented at the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV).

Iacono, J., Brown, A., & Holtham, C. (2009). Research Methods--a Case Example of Participant Observation. *Electronic journal of business research methods, 7*(1).

Iivari, J. (1990). Hierarchical spiral model for information system and software development. Part 2: Design process. *Information and Software Technology, 32*(7), 450-458.

Ishizaka, A., & Siraj, S. (2018). Are multi-criteria decision-making tools useful? An experimental comparative study of three methods. *European Journal of Operational Research, 264*(2), 462-471.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., . . . Kalenichenko, D. (2018). *Quantization and training of neural networks for efficient integer-arithmetic-only inference.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Jagadeesh, V., Piramuthu, R., Bhardwaj, A., Di, W., & Sundaresan, N. (2014). *Large scale visual recommendations from street fashion images.* Paper presented at the Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.

James III, R. (2009). Customer satisfaction with apartment housing offered by Real Estate Investment Trusts (REITs). *International Journal of Consumer Studies, 33*(5), 572-580.

Jancsary, J., Nowozin, S., & Rother, C. (2012). *Loss-specific training of non-parametric image restoration models: A new state of the art.* Paper presented at the European Conference on Computer Vision.

Jarosz, M., Kutrzyński, M., Lasota, T., Piwowarczyk, M., Telec, Z., & Trawiński, B. (2020). *Machine Learning Models for Real Estate Appraisal Constructed Using Spline Trend Functions.* Paper presented at the Asian Conference on Intelligent Information and Database Systems.

Jiang, Z., Chan, J., Tan, B. C., & Chua, W. S. (2010). Effects of interactivity on website involvement and purchase intention. *Journal of the Association for Information Systems, 11*(1), 1.

Jun, M., & Palacios, S. (2016). Examining the key dimensions of mobile banking service quality: an exploratory study. *International Journal of Bank Marketing*.

Kaufman, J. G. (1991). Increasing data system responsiveness to end-user expectations. In *Computerization and Networking of Materials Databases: Second Volume*: ASTM International.

Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing* (pp. 187-219): Springer.

Kinsner, W. (2009). Challenges in the Design of Adoptive, Intelligent and Cognitive Systems. *International Journal of Software Science and Computational Intelligence (IJSSCI), 1*(3), 16-35.

Kothari, C. R. (2004). *Research methodology: Methods and techniques*: New Age International.

Krishnakumar, K. (2003). Intelligent systems for aerospace engineering-an overview.

Kuechler, W., & Vaishnavi, V. (2012). A framework for theory development in design science research: multiple perspectives. *Journal of the Association for Information systems, 13*(6), 3.

Kuhn, T. S. (1970). *Criticism and the growth of knowledge: Volume 4: Proceedings of the International Colloquium in the Philosophy of Science, London, 1965* (Vol. 4): Cambridge University Press.

Kumar, A., Sah, B., Singh, A. R., Deng, Y., He, X., Kumar, P., & Bansal, R. (2017). A review of multi criteria decision making (MCDM) towards sustainable renewable energy development. *Renewable and Sustainable Energy Reviews, 69*, 596-609.

Kumar, E. S., Talasila, V., Rishe, N., Kumar, T. S., & Iyengar, S. (2019). Location identification for real estate investment using data analytics. *International Journal of Data Science and Analytics, 8*(3), 299-323.

Kumar, R., Mukherjee, A., & Singh, V. (2017). Community sensor network for monitoring road roughness using smartphones. *Journal of Computing in Civil Engineering, 31*(3), 04016059.

Kwon, W., Yu, G.-I., Jeong, E., & Chun, B.-G. (2020). Nimble: Lightweight and parallel gpu task scheduling for deep learning. *arXiv preprint arXiv:2012.02732*.

Lebib, F. Z., Drias, H., & Mellah, H. (2017). *Selection of information sources using a genetic algorithm.* Paper presented at the World Conference on Information Systems and Technologies.

LeCun, Y., Bengio, Y., & Hinton, G. (2015a). Deep learning. *nature, 521*(7553), 436-444.

Lee, E.-K., Gerla, M., Pau, G., Lee, U., & Lim, J.-H. (2016). Internet of Vehicles: From intelligent grid to autonomous cars and vehicular fogs. *International Journal of Distributed Sensor Networks, 12*(9), 1550147716665500.

Lee, Y. S. (2018). Watching You Watching Me: Lateral Surveillance in Singapore.

Lee, Z., Chan, T., Balaji, M., & Chong, A. Y.-L. (2016). *Technology-mediated sharing economy: Understanding user participation in collaborative consumption through the benefit-cost perspective.* Paper presented at the Proceedings of the 20th Pacific Asia Conference on Information Systems (PACIS).

Leminen, S., Rajahonka, M., Wendelin, R., & Westerlund, M. (2020). Industrial internet of things business models in the machine-to-machine context. *Industrial Marketing Management, 84*, 298-311.

Levinger, C., Hazon, N., & Azaria, A. (2020). Human satisfaction as the ultimate goal in ridesharing. *Future Generation Computer Systems*.

Li, S., Da Xu, L., & Zhao, S. (2018). 5G Internet of Things: A survey. *Journal of Industrial Information Integration, 10*, 1-9.

Li, Y., & Taeihagh, A. The governance of risks in ridesharing: Lessons learned from Singapore.

Limsombunchai, V. (2004). *House price prediction: hedonic price model vs. artificial neural network.* Paper presented at the New Zealand agricultural and resource economics society conference.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). *Feature pyramid networks for object detection.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Liu, L., Zhang, X., Qiao, M., & Shi, W. (2018). *SafeShareRide: Edge-based attack detection in ridesharing services.* Paper presented at the 2018 IEEE/ACM Symposium on Edge Computing (SEC).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). *Ssd: Single shot multibox detector.* Paper presented at the European conference on computer vision.

Loiacono, E. T., Watson, R. T., & Goodhue, D. L. (2002). WebQual: A measure of website quality. *Marketing theory and applications, 13*(3), 432-438.

Long, C., Cao, Y., Jiang, T., & Zhang, Q. (2017). Edge computing framework for cooperative video processing in multimedia IoT systems. *IEEE Transactions on Multimedia, 20*(5), 1126-1139.

Lu, C., Kong, Y., & Guan, Z. (2020). A mask R-CNN model for reidentifying extratropical cyclones based on quasi-supervised thought. *Scientific Reports, 10*(1), 1-9.

Lunardi, W. T. (2016). Decision support IoT framework: device discovery and stream analytics.

Lyft. (2019). Code of Business Conduct and Ethics. Retrieved from https://investor.lyft.com/static-files/8861ea0b-6a32-4414-a909-d8a9618f582f

Ma, G., Wang, Z., Zhang, M., Ye, J., Chen, M., & Zhu, W. (2017). Understanding performance of edge content caching for mobile video streaming. *IEEE Journal on Selected Areas in Communications, 35*(5), 1076-1089.

Macmurdo, M. (2015). Hold the Phone-Peer-to-Peer Ridesharing Services, Regulation, and Liability. *La. L. Rev., 76*, 307.

Margaria, T., & Steffen, B. (2006). Service engineering: Linking business and it. *Computer, 39*(10), 45-55.

Marshall, T. E., & Lambert, S. L. (2018). Cloud-based intelligent accounting applications: accounting task automation using IBM watson cognitive computing. *Journal of emerging technologies in accounting, 15*(1), 199-215.

McCord, M. J., McCord, J., Davis, P. T., Haran, M., & Bidanset, P. (2019). House price estimation using an eigenvector spatial filtering approach. *International Journal of Housing Markets and Analysis*.

Meshram, A., Choudhary, P., & Velaga, N. R. (2020). Assessing and modelling perceived safety and comfort of women during ridesharing. *Transportation research procedia, 48*, 2852-2869.

Michael, M., Fusco, S. J., & Michael, K. (2008). A research note on ethics in the emerging age of überveillance. *Computer communications, 31*(6), 1192-1199.

Miller, T. R., Courser, M., Shamblen, S. R., Lange, J. E., Tippetts, A. S., & Ringwalt, C. (2020). Efficacy and cost-effectiveness of subsidized ridesharing as a drunk driving intervention in Columbus, OH. *Accident Analysis & Prevention, 146*, 105740.

Mohamadnejad, M., Gholami, R., & Ataei, M. (2012). Comparison of intelligence science techniques and empirical methods for prediction of blasting vibrations. *Tunnelling and Underground Space Technology, 28*, 238-244.

Mousa, A., & Saadeh, M. (2010). Automatic valuation of Jordanian estates using a genetically-optimised artificial neural network approach. *WSEAS Transactions on Systems, 9*, 905-916.

Munteanu, C., & Lazarescu, V. (1999). *Evolutionary contrast stretching and detail enhancement of satellite images.* Paper presented at the Proc. Mendel.

Murakami, D., & Griffith, D. A. (2019). Spatially varying coefficient modeling for large datasets: Eliminating N from spatial regressions. *Spatial Statistics, 30*, 39-64.

Newman, I., Benz, C. R., & Ridenour, C. S. (1998). *Qualitative-quantitative research methodology: Exploring the interactive continuum*: SIU Press.

Ng, P. C., She, J., Cheung, M., & Cebulla, A. (2016). *An images-textual hybrid recommender system for vacation rental.* Paper presented at the 2016 IEEE Second International Conference on Multimedia Big Data (BigMM).

Nie, X., Duan, M., Ding, H., Hu, B., & Wong, E. K. (2020). Attention Mask R-CNN for ship detection and segmentation from remote sensing images. *IEEE Access, 8*, 9325-9334.

O'Connor, Y., Rowan, W., Lynch, L., & Heavin, C. (2017). Privacy by design: informed consent and internet of things for smart health. *Procedia computer science, 113*, 653-658.

Osei-Bryson, K.-M. (2004). Evaluation of decision trees: a multi-criteria approach. *Computers & Operations Research, 31*(11), 1933-1945.

Pakurár, M., Haddad, H., Nagy, J., Popp, J., & Oláh, J. (2019). The service quality dimensions that affect customer satisfaction in the Jordanian banking sector. *Sustainability, 11*(4), 1113.

Palvia, P., Mao, E., Salam, A., & Soliman, K. S. (2003). Management information systems research: what's there in a methodology? *Communications of the Association for Information Systems, 11*(1), 16.

Panda, T. K., & Das, S. (2014). The role of tangibility in service quality and its impact on external customer satisfaction: A comparative study of hospital and hospitality sectors. *IUP Journal of Marketing Management, 13*(4), 53.

Parasuraman, A., Zeithaml, V. A., & Malhotra, A. (2005). ES-QUAL: A multiple-item scale for assessing electronic service quality. *Journal of service research, 7*(3), 213-233.

Peffers, K., Rothenberger, M., Tuunanen, T., & Vaezi, R. (2012). *Design science research evaluation.* Paper presented at the International Conference on Design Science Research in Information Systems.

Poursaeed, O., Matera, T., & Belongie, S. (2018). Vision-based real estate price estimation. *Machine Vision and Applications, 29*(4), 667-676.

Pupkov, K. (2019). Intelligent systems and human being. *Procedia Computer Science, 150*, 540-543.

Pur, P. (2017). Effects of flexibility and interactivity on the perceived value of and satisfaction with e-commerce (evidence from Indonesia). *Market-Trziste, 29*(2), 139-159.

Rabiei-Dastjerdi, H., McArdle, G., Matthews, S. A., & Keenan, P. (2021). Gap analysis in decision support systems for real-estate in the era of the digital earth. *International Journal of Digital Earth, 14*(1), 121-138.

Ragini, J. R., Anand, P. R., & Bhaskar, V. (2018). Big data analytics for disaster response and recovery through sentiment analysis. *International Journal of Information Management, 42*, 13-24.

Ran, X., Chen, H., Zhu, X., Liu, Z., & Chen, J. (2018). *Deepdecision: A mobile deep learning framework for edge video analytics.* Paper presented at the IEEE INFOCOM 2018-IEEE Conference on Computer Communications.

Ransbotham, S., Kiron, D., Gerbert, P., & Reeves, M. (2017). Reshaping business with artificial intelligence: Closing the gap between ambition and action. *MIT Sloan Management Review, 59*(1).

Rekik, R., Kallel, I., Casillas, J., & Alimi, A. M. (2018). Assessing web sites quality: A systematic literature review by text and association rules mining. *International Journal of Information Management, 38*(1), 201-216.

Rubin, A., & Babbie, E. R. (2012). *Brooks/Cole Empowerment Series: Essential research methods for social work*: Cengage Learning.

Saaty, T. L., & Vargas, L. G. (1984). Inconsistency and rank preservation. *Journal of Mathematical Psychology, 28*(2), 205-214.

Samah, K., Badarudin, I., Odzaly, E., Ismail, K., Nasarudin, N., Tahar, N., & Khairuddin, M. (2019). Optimization of house purchase recommendation system (HPRS) using genetic algorithm. *Indonesian Journal of Electrical Engineering and Computer Science, 16*(3), 1530-1538.

Santos, J. (2002). From intangibility to tangibility on service quality perceptions: a comparison study between consumers and service providers in four service industries. *Managing Service Quality: An International Journal*.

Santos, J. (2003). E-service quality: a model of virtual service quality dimensions. *Managing Service Quality: An International Journal*.

Satapathy, S. M., Jhaveri, R., Khanna, U., & Dwivedi, A. K. (2020). Smart Rent Portal using Recommendation System Visualized by Augmented Reality. *Procedia Computer Science, 171*, 197-206.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks, 61*, 85-117.

Schreyer, M. (2018). *Applying Recommender Approaches to the Real Estate e-Commerce Market.* Paper presented at the Innovations for Community Services: 18th International Conference, I4CS 2018, Žilina, Slovakia, June 18-20, 2018, Proceedings.

Seiler, V., & Reisenwitz, T. (2010). A review of service quality research in real estate. *Journal of real estate literature, 18*(2), 225-238.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action design research. *MIS quarterly*, 37-56.

Senior, A. (2009). Privacy protection in a video surveillance system. In *Protecting Privacy in Video Surveillance* (pp. 35-47): Springer.

Sharma, G., & Lijuan, W. (2015). The effects of online service quality of e-commerce Websites on user satisfaction. *The Electronic Library*.

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal, 3*(5), 637-646.

Shwartz-Ziv, R., & Tishby, N. (2017). Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

Singh, M., & Baranwal, G. (2018). *Quality of service (qos) in internet of things.* Paper presented at the 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU).

Singh, P., Manikandan, R., Matiyali, N., & Namboodiri, V. P. (2019). *Multi-layer pruning framework for compressing single shot multibox detector.* Paper presented at the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV).

Soman, S., & Suri, M. (2016a). Recent trends in neuromorphic engineering. *Big Data Analytics, 1*(1), 1-19.

Sperandio, F., Gomes, C., Borges, J., Brito, A. C., & Almada-Lobo, B. (2013). An intelligent decision support system for the operating theater: A case study. *IEEE transactions on automation science and engineering, 11*(1), 265-273.

Sugumaran, V. (1998). A Distributed Intelligent Agent-Based Spatial Decision Support System. *AMCIS 1998 Proceedings*, 136.

Sun, H., Teh, P.-L., & Chiu, A. (2012). An empirical study on the websites service quality of Hong Kong small businesses. *Total Quality Management & Business Excellence, 23*(7-8), 931-947.

Synced. (2017). Deep Learning in Real Time – Inference Acceleration and Continuous Training.

Tekile, H. A., Fedrizzi, M., & Brunelli, M. (2021). Constrained Eigenvalue Minimization of Incomplete Pairwise Comparison Matrices by Nelder-Mead Algorithm. *Algorithms, 14*(8), 222.

Ullah, F., & Sepasgozar, S. (2019). A study of information technology adoption for real-estate management: A system dynamic model. *Innovative Production And Construction*, 469-486.

Ullah, F., Sepasgozar, S. M., & Wang, C. (2018). A systematic review of smart real estate technology: Drivers of, and barriers to, the use of digital disruptive technologies and online platforms. *Sustainability, 10*(9), 3142.

Venable, J. (2006). *The role of theory and theorising in design science research.* Paper presented at the Proceedings of the 1st International Conference on Design Science in Information Systems and Technology (DESRIST 2006).

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science, 46*(2), 186-204.

Wang, L., Zhang, Q., Li, Y., Zhong, H., & Shi, W. (2019). *MobileEdge: Enhancing On-Board Vehicle Computing Units Using Mobile Edges for CAVs.* Paper presented at the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS).

Wang, T. (2020). *Application of Edge Computing in 5g Communications.* Paper presented at the IOP Conference Series: Materials Science and Engineering.

Wang, Y., Anderson, J. A., Baciu, G., Budin, G., Hsu, D. F., Ishizuka, M., . . . Sugawara, K. (2012). Perspectives on eBrain and cognitive computing. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), 6*(4), 1-21.

Weber, B. S. (2019). Uber and urban crime. *Transportation research part A: policy and practice, 130*, 496-506.

West, S. M. (2019). Data capitalism: Redefining the logics of surveillance and privacy. *Business & society, 58*(1), 20-41.

Wikipedia. (2019). Deep Learning. Retrieved from http://en.wikipedia.org/wiki/Deep_learning

Winson-Geideman, K., & Krause, A. (2016). *Transformations in real estate research: The big data revolution.* Paper presented at the Presentation at the 22nd Annual Pacific-Rim Real Estate Society Conference.

Wolfinbarger, M., & Gilly, M. C. (2003). eTailQ: dimensionalizing, measuring and predicting etail quality. *Journal of retailing, 79*(3), 183-198.

Wu, J., Zhang, Y., Bai, H., Zhong, H., Hou, J., Liu, W., . . . Huang, J. (2018). PocketFlow: An Automated Framework for Compressing and Accelerating Deep Neural Networks.

Wu, N., Gelman, I. A., & Osesina, I. O. (2009). How consistent is web information-A case study on online real estate databases. *AMCIS 2009 Proceedings*, 437.

Xu, J., Benbasat, I., & Cenfetelli, R. T. (2013). Integrating service quality with system and information quality: an empirical test in the e-service context. *MIS quarterly*, 777-794.

Xue, M., Harker, P. T., & Heim, G. R. (2000). Website efficiency, customer satisfaction and customer loyalty: A customer value driven perspective. *The Wharton School University of Pennsylvania*.

Yadav, M. S., & Varadarajan, R. (2005). Interactivity in the electronic marketplace: An exposition of the concept and implications for research. *Journal of the Academy of Marketing Science, 33*(4), 585-603.

Yaqoob, I., Hashem, I. A. T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N. B., & Vasilakos, A. V. (2016). Big data: From beginning to future. *International Journal of Information Management, 36*(6), 1231-1247.

Yi, S., Hao, Z., Zhang, Q., Zhang, Q., Shi, W., & Li, Q. (2017). *Lavea: Latency-aware video analytics on edge computing platform.* Paper presented at the Proceedings of the Second ACM/IEEE Symposium on Edge Computing.

You, Q., Pang, R., Cao, L., & Luo, J. (2017). Image-based appraisal of real estate properties. *IEEE transactions on multimedia, 19*(12), 2751-2759.

Yuan, X., Lee, J.-H., Kim, S.-J., & Kim, Y.-H. (2013). Toward a user-oriented recommendation system for real estate websites. *Information Systems, 38*(2), 231-243.

Zantout, H., & Marir, F. (1999). Document management systems from current capabilities towards intelligent information retrieval: an overview. *International Journal of Information Management, 19*(6), 471-484.

Zeithaml, V. A., Parasuraman, A., & Malhotra, A. (2002). Service quality delivery through web sites: a critical review of extant knowledge. *Journal of the academy of marketing science, 30*(4), 362-375.

Zhang, Q., Zha, L., Wan, X., & Cheng, B. (2019). *A GPU Inference System Scheduling Algorithm with Asynchronous Data Transfer.* Paper presented at the 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW).

Zhang, X., Chen, F., & Huang, R. (2018). A combination of RNN and CNN for attention-based relation classification. *Procedia Computer Science, 131*, 911-917.

Zhang, X., Wang, Y., Lu, S., Liu, L., & Shi, W. (2019). *Openei: An open framework for edge intelligence.* Paper presented at the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS).

Zhao, L., Lu, Y., Zhang, L., & Chau, P. Y. (2012). Assessing the effects of service quality and justice on customer satisfaction and the continuance intention of mobile value-added services: An empirical test of a multidimensional model. *Decision support systems, 52*(3), 645-656.

Zhao, Y., Tang, L. C., Darlington, M. J., Austin, S. A., & Culley, S. J. (2008). High value information in engineering organisations. *International Journal of Information Management, 28*(4), 246-258.

Zhaohao, S. (2019). Intelligent Big Data Analytics: A Managerial Perspective. In S. Zhaohao (Ed.), *Managerial Perspectives on Intelligent Big Data Analytics* (pp. 1-19). Hershey, PA, USA: IGI Global.

Zhaohao, S., Ping, Z., & Dong, D. (2012). Customer Decision Making in Web Services. In K. Ejub, D. Frank-Dieter, & B. Ivan (Eds.), *Handbook of Research on E-Business Standards and Protocols: Documents, Data and Advanced Web Technologies* (pp. 210-232). Hershey, PA, USA: IGI Global.

Zopounidis, C., & Doumpos, M. (2002). Multi-criteria decision aid in financial decision making: methodologies and literature review. *Journal of Multi-Criteria Decision Analysis, 11*(4-5), 167-186.