
Electronic Theses and Dissertations, 2004-2019

2015

Data-Driven Simulation Modeling of Construction and Infrastructure Operations Using Process Knowledge Discovery

Reza Akhavian

University of Central Florida, reza.akhavian@csueastbay.edu

 Part of the [Civil Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd>

University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Akhavian, Reza, "Data-Driven Simulation Modeling of Construction and Infrastructure Operations Using Process Knowledge Discovery" (2015). *Electronic Theses and Dissertations, 2004-2019*. 1419.

<https://stars.library.ucf.edu/etd/1419>

DATA-DRIVEN SIMULATION MODELING OF CONSTRUCTION AND
INFRASTRUCTURE OPERATIONS USING PROCESS KNOWLEDGE DISCOVERY

by

REZA AKHAVIAN
B.S. University of Tehran, 2010
M.S.C.E. University of Central Florida, 2012

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Civil, Environmental, and Construction Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2015

Major Professor: Amir H. Behzadan

© 2015 Reza Akhavian

ABSTRACT

Within the architecture, engineering, and construction (AEC) domain, simulation modeling is mainly used to facilitate decision-making by enabling the assessment of different operational plans and resource arrangements, that are otherwise difficult (if not impossible), expensive, or time consuming to be evaluated in real world settings. The accuracy of such models directly affects their reliability to serve as a basis for important decisions such as project completion time estimation and resource allocation. Compared to other industries, this is particularly important in construction and infrastructure projects due to the high resource costs and the societal impacts of these projects. Discrete event simulation (DES) is a decision making tool that can benefit the process of design, control, and management of construction operations. Despite recent advancements, most DES models used in construction are created during the early planning and design stage when the lack of factual information from the project prohibits the use of realistic data in simulation modeling. The resulting models, therefore, are often built using rigid (subjective) assumptions and design parameters (e.g. precedence logic, activity durations). In all such cases and in the absence of an inclusive methodology to incorporate real field data as the project evolves, modelers rely on information from previous projects (a.k.a. secondary data), expert judgments, and subjective assumptions to generate simulations to predict future performance. These and similar shortcomings have to a large extent limited the use of traditional DES tools to preliminary studies and long-term planning of construction projects.

In the realm of the business process management, process mining as a relatively new research domain seeks to automatically discover a process model by observing activity records and extracting information about processes. The research presented in this Ph.D. Dissertation was in part inspired by the prospect of construction process mining using sensory data collected from field agents. This enabled the extraction of operational knowledge necessary to generate and maintain the fidelity of simulation models. A preliminary study was conducted to demonstrate the feasibility and applicability of data-driven knowledge-based simulation modeling with focus on data collection using wireless sensor network (WSN) and rule-based taxonomy of activities. The resulting knowledge-based simulation models performed very well in properly predicting key performance measures of real construction systems. Next, a pervasive mobile data collection and mining technique was adopted and an activity recognition framework for construction equipment and worker tasks was developed. Data was collected using smartphone accelerometers and gyroscopes from construction entities to generate significant statistical time- and frequency-domain features. The extracted features served as the input of different types of machine learning algorithms that were applied to various construction activities. The trained predictive algorithms were then used to extract activity durations and calculate probability distributions to be fused into corresponding DES models. Results indicated that the generated data-driven knowledge-based simulation models outperform static models created based upon engineering assumptions and estimations with regard to compatibility of performance measure outputs to reality.

To my lovely parents
For all your support and encouragements

ACKNOWLEDGMENTS

I am truly grateful to my Ph.D. advisor, Dr. Amir H. Behzadan, who opened the door of real research to me in 2010 when I started my Master's studies under his supervision. His support, patience, wisdom, and friendship made the challenging journey of graduate studies a very special and enjoyable one for me. I will be always thankful for his trust in me to explore various opportunities and for all I learned from him in my professional career and life.

I would like to thank my other committee members, Dr. Oloufa, Dr. Yun, Dr. Sukthankar, and Dr. Zheng for their advice and constructive input. Specifically, I would like to thank Dr. Oloufa who always offered support and guidance to me during the last four years.

I am very thankful to many of my friends and colleagues for their sincere friendship and help. Specifically Behnaz Seyfi, Karim Alizad, Sona Gholizadeh, Siamak Ebadi, Shadi Aslebagh, and Alireza Noorollahi for being there in life hardships, and the members of construction research laboratory at the University of Central Florida for helping me with my research experiments. Special thanks go out to my wonderful friends in the Saman Music Ensemble based in Orlando, FL, USA.

My deepest gratitude is due to my family; my lovely parents Hossein Akhavian and Maryam Batebi for their unconditional love and support, and my one and only brother Armin. Thanks for being my source of motivation and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES	xii
LIST OF TABLES.....	xv
CHAPTER 1: INTRODUCTION.....	xv
1.1 Background and Problem Statement.....	1
1.2 Research Objectives.....	2
1.3 Research Contribution.....	5
1.4 Research Methodology.....	5
1.5 Organization of the Dissertation	7
CHAPTER 2: KNOWLEDGE-BASED SIMULATION MODELING OF CONSTRUCTION FLEET OPERATIONS USING MULTI-MODAL PROCESS DATA MINING	10
2.1 Introduction	10
2.2 Main Contributions to the Body of Knowledge.....	12
2.3 Methodology	13
2.3.1 Multi-Modal Data Collection and Fusion.....	13
2.3.2 Knowledge Discovery and Reasoning Process.....	17
2.3.3 K-Means Clustering.....	20
2.3.4 Contextual Knowledge Discovery Using K-Means.....	21

2.3.5	Automated Simulation Model Generation	25
2.4	Results	27
2.4.1	Experiment 1: One Loader and One Dump Truck, Without Service Area	28
2.4.2	Experiment 2: One Loader and Multiple Dump Trucks, With Service Area	31
2.4.3	Data-Driven Simulation	34
2.5	Summary and Conclusions	37
CHAPTER 3: EVALUATION OF QUEUING SYSTEMS FOR KNOWLEDGE-BASED SIMULATION OF CONSTRUCTION PROCESSES.....		39
3.1	Introduction	39
3.2	Research Background.....	43
3.3	Queue Properties	45
3.3.1	The arrival process	46
3.3.2	The service process	46
3.3.3	The queue discipline	47
3.4	Mathematical Representation of Interarrival and Service Times.....	49
3.5	Queue Property Discovery Algorithms	53
3.5.1	Interarrival and service times.....	53
3.5.2	Queue disciplines	54
3.6	Empirical Analysis and Validation of Results	62

3.7	Summary and Conclusions.....	76
CHAPTER 4: CONSTRUCTION EQUIPMENT ACTIVITY RECOGNITION FOR SIMULATION INPUT MODELING USING MOBILE SENSORS AND MACHINE LEARNING CLASSIFIERS		
4.1	Introduction	79
4.2	Previous Work.....	81
4.2.1	Action Recognition Using Accelerometer and Gyroscope Data	82
4.2.2	Construction Resource Action Recognition.....	83
4.2.3	Simulation Modeling in Construction.....	86
4.3	Level of Detail in Equipment Activity Recognition	87
4.4	Activity Recognition Methodology.....	92
4.4.1	Experiment Setup.....	94
4.4.2	Data Collection and Logging	95
4.4.3	Data Processing.....	96
4.5	Results and Discussion.....	100
4.6	Summary and Conclusions.....	108
CHAPTER 5: SMARTPHONE-BASED ACTIVITY RECOGNITION AND CLASSIFICATION OF CONSTRUCTION WORKERS.....		
5.1	Introduction	111

5.2	Literature Review	113
5.2.1	Automated recognition of construction worker activities.....	113
5.2.2	Activity recognition using cellphone sensors	115
5.3	Research Objectives and Contributions to the Body of Knowledge.....	116
5.4	Methodology	117
5.4.1	Data Acquisition Using Mobile Phones.....	117
5.4.2	Data Preparation.....	118
5.4.3	Feature Extraction.....	119
5.4.4	Data Annotation	119
5.4.5	Supervised Learning	119
5.4.6	Model Assessment	120
5.4.7	Activity Recognition.....	120
5.5	Experiments Setup and Data Analysis	120
5.5.1	Data Collection	121
5.5.2	Data Analysis.....	122
5.5.3	Classifier Training	125
5.6	Results and Discussion.....	129
5.7	Summary and Conclusions.....	138
5.8	Future Work	140

CHAPTER 6: DATA-DRIVEN SIMULATION OF CONSTRUCTION PROCESSES WITH COMPLEX WORKER INTERACTIONS USING SMARTPHONE SENSOR-BASED ACTIVITY RECOGNITION	143
6.1 Introduction	143
6.2 The Operation Experiment Design.....	144
6.2.1 Experiment Setup.....	144
6.2.2 Assumptions and Rules.....	146
6.3 Simulation Model of the Operation.....	147
6.4 Duration Extraction through Activity Recognition.....	150
6.5 Simulation Input Modeling	154
6.6 Performance of the Data-Driven Model vs. Static Simulation Model	157
6.7 Discussion of the Results	159
6.8 Summary and Conclusions.....	160
CHAPTER 7: CONCLUSIONS	162
APPENDIX A: COPYRIGHT PERMISSION	167
APPENDIX B: AUTHOR’S BIOGRAPHY	169
REFERENCES	171

LIST OF FIGURES

Figure 2.1: Taxonomy-based state classification of construction fleet.....	18
Figure 2.2: Taxonomy of dump truck activities in an earthmoving operation based upon multi-modal process data and operational context.	23
Figure 2.3: Layout of laboratory experiments.	28
Figure 2.4: Results of K-means clustering algorithm applied to the positional and weight data in 2D (XY) and 3D (XYW) spaces for experiment 1.	29
Figure 2.5: Detecting changes in the location of cluster centroids over time.....	31
Figure 2.6: Results of K-means clustering algorithm applied to the positional and weight data in 2D (XY) and 3D (XYW) spaces for Experiment 2.....	32
Figure 2.7: ACD of the DES model of a typical earthmoving operation.	35
Figure 2.8: Analysis of results obtained from the real-world experiment and the output of traditional and data-driven simulation models.....	36
Figure 3.1: Example of a single server queuing system in construction projects.....	40
Figure 3.2: Demonstration of FIFO, LIFO, and PRI queue disciplines.....	48
Figure 3.3: Sequence and the properties of a queuing system.....	48
Figure 3.4: Main flowchart for queue discipline discovery.....	55
Figure 3.5: FIFO queue discipline discovery flowchart.	56
Figure 3.6: LIFO queue discipline discovery flowchart.	58

Figure 3.7: Assigning priorities to clients in a PRI queuing system.....	60
Figure 3.8: PRI queue discipline discovery flowchart.....	61
Figure 3.9: Schematic illustration of the experiment test bed.	64
Figure 3.10: Gamma Distribution Fitted to the Extracted Interarrival Times in Experiment 1....	68
Figure 3.11: CDF of the experiment input data and the best fit distribution to the extracted durations.....	70
Figure 3.12: Activity cycle diagram of the queuing system in Stroboscope.	73
Figure 3.13: Stroboscope simulation script and the implementation of the discovered queue property change.....	74
Figure 3.14: Comparison of average waiting times of even and odd ID tags.....	76
Figure 4.1: LoD in activity breakdown of a front-end loader.....	89
Figure 4.2: Developed system architecture for simulation input modeling.....	93
Figure 4.3: Smartphones mounted inside the front-end loader cabin.	94
Figure 4.4: Snapshots of collected sensory data.	96
Figure 4.5: Neural network confusion matrix for level 2 (A: Engine off, B: Idle, C: Busy).....	104
Figure 4.6: Neural network confusion matrix for level 3 (A: Engine off, B: Idle, C: Moving and Scooping, D: Moving and Dumping).....	104
Figure 4.7: Neural network confusion matrix for level 4 (A: Engine off, B: Idle, C: Scooping, D: Moving, E: Dumping).....	105
Figure 5.1: Framework for construction worker activity recognition using mobile sensors.....	117
Figure 5.2: Data collection experiments (mobile devices are marked with dashed circles).....	121
Figure 5.3: The architecture of the neural network used in this research.	125

Figure 5.4: Confusion matrices of 10-fold cross validation of neural network classification for category 1 activities.	131
Figure 5.5: Confusion matrices of 10-fold cross validation of neural network classification for category 2 activities.	132
Figure 5.6: Confusion matrices of 10-fold cross validation of neural network classification for category 3 activities.	134
Figure 5.7: Confusion matrices of 10-fold cross validation of neural network classification for combined data of subjects I and II in all three activity categories.....	137
Figure 5.8: Discovered time allocation proportions in the conducted experiments, for productivity measurement.....	141
Figure 6.1: Schematic illustration of the operation experiment setup.	145
Figure 6.2: Snapshot of the operation showing four workers performing the experiment.	145
Figure 6.3: The ACD of the operation for modeling in Stroboscope.	148
Figure 6.4: Comparison of results obtained from the real-world experiment, and the output of static and data-driven simulation models.....	158
Figure A.1: ASCE permission to reuse the published paper in Chapter 2 of this Dissertation. .	168

LIST OF TABLES

Table 2.1: Specifications of employed sensors in the distributed network.....	27
Table 2.2: Centroids of the detected clusters using K-means for experiment 1.	29
Table 2.3: Observed vs. extracted activity duration means and standard deviations for experiment 1.....	30
Table 2.4: Centroids of the detected clusters using K-means for experiment 2.	32
Table 2.5: Observed vs. extracted activity duration means and standard deviations for experiment 2.....	33
Table 2.6: Observed vs. extracted means and standard deviations of waiting times for experiment 2.....	34
Table 2.7: Approximated activity durations based on overall site layout and resource specifications.....	35
Table 3.1: Sample of the collected data points.	63
Table 3.2: Sample of the OA, OS, interarrival, and service times extracted in Experiment 1.	66
Table 3.3: Ranking of the Best fitted probability distributions for Experiment 1.	67
Table 3.4: Best fit distribution for Experiments 1, 2, and 3.....	69
Table 3.5: Results of the queue discipline discovery.....	71
Table 3.6: Input parameters of the simulation model for Experiment 4.	73
Table 4.1: Selected features by CFS and ReliefF and their intersection (A: Accelerometer, G: Gyroscope).....	98
Table 4.2: Overall accuracy of classifiers for each LoD.	101

Table 4.3: Accuracy of the trained neural networks for different window sizes.....	102
Table 4.4: Number of segments and instances for each activity within each class.	103
Table 4.5: NRMSEs of extracted activity durations compared to the real activity durations. ...	106
Table 5.1: Collected data points per sensor per axis in each activity category.	123
Table 5.2: Classification accuracy (%) for category 1 activities.	130
Table 5.3: Classification accuracy (%) for category 2 activities.	131
Table 5.4: Classification accuracy (%) for category 3 activities.	133
Table 5.5: Classification accuracy (%) for combined data of subjects I and II in all three activity categories.	135
Table 5.6: Accuracy (%) of classifiers trained with data from one subject and tested on data from another subject.	138
Table 6.1: List of the processes involved in the operation and activities within each process...	146
Table 6.2: Sensory data collection configurations used for activity recognition.....	151
Table 6.3: Neural network Bagging classification accuracy (%) for recognizing activities within each process.	152
Table 6.4: Comparison of the observed and extracted activity durations using student t-test....	155
Table 6.5: Probability distributions used inside the two simulation models.	156

CHAPTER 1: INTRODUCTION

1.1 Background and Problem Statement

Discrete event simulation (DES) is a powerful objective function evaluator that can benefit the process of design, control, and management of construction operations [1, 2, 3, 4]. Most construction and infrastructure projects consist of discrete activities or sub-systems which make them ideal for DES modeling. A DES model is an event-based representation of project activities that constitute an engineering system. In order to develop a realistic simulation model, it is critical to provide the model with factual input data based on the interactions and events that take place between real entities. However, the existing trend in simulation of construction activities is based on estimating input parameters such as activity durations using expert judgments and assumptions. Not only such estimations may not be precise, project dynamics can influence model parameters beyond expectation. Therefore, the simulation model may not be a proper and reliable representation of the real engineering system. In order to alleviate these issues and improve the current practice of construction simulation, a thorough approach is needed that enables the integration of field data into simulation modeling and systematic refinement of the resulting models. Moreover, during the course of a construction project, there are many situations in which formation of waiting-lines or queues is inevitable. The effect of resource delays in queues on the overall project completion time and cost has motivated researchers to

employ simulation for analysis of queuing systems in order to identify the best operational strategies to reduce the time wasted in queues. Providing proper and timely input data with high spatial and temporal accuracy for queuing systems simulation enhances the reliability of decisions made based upon the simulation output.

Data collection and knowledge extraction are essential components of a framework aimed at updating and fine-tuning simulation models. Collecting data for activity recognition through pervasive devices equipped with sensors such as mobile phones is an emerging computer science research area. Similarly, in the realm of the business process management, process mining as a relatively new research domain seeks to automatically discover a process model by observing activity records and extracting information about processes. However, the potential of these two emerging general areas of research has not yet been fully investigated in construction engineering and management (CEM). Due to the complex and dynamic nature of many construction and infrastructure projects, the ability to detect and classify key activities performed in the field by various equipment and crews can improve the quality and reliability of project decision-making and control. In particular to simulation modeling, process-level knowledge obtained as a result of activity recognition can help verify and update the input parameters of simulation models. Such input parameters include but are not limited to activity durations and precedence, resource flows, and site layout.

1.2 Research Objectives

The goal of this doctoral research is to investigate the design and implementation of a framework capable of capturing multi-modal process data from field agents (i.e. construction resources)

using ubiquitous mobile sensors, and extracting computer-interpretable knowledge through vigorous data mining and machine learning techniques to update DES models corresponding to the real system operations. To achieve this goal, the following research objectives were identified and research questions, the hypotheses investigated to answer them, and the significance of associated research tasks were outlined:

- **Objective 1:** To increase the accuracy of construction simulation models using data collected from the field agents and extracting factual knowledge.
 - **Question:** How can a data-driven simulation framework enhance the accuracy and quality of simulation modeling for construction and infrastructure projects? How can it contribute towards creating more realistic simulation results?
 - **Hypothesis:** Collecting data from construction equipment and workers helps in constructing a rule-based taxonomy to determine realistic activity durations and approximate the layout of a construction jobsite.
 - **Significance:** The answer to this question sheds light on the applicability and feasibility of using data-driven knowledge-based construction simulation models.
- **Objective 2:** To investigate the interaction between clients waiting in queues with the servers giving service to clients inside a construction jobsite.
 - **Question:** What is the procedure to extract queue properties from data collected from clients and servers? What modes of data are required to achieve this objective? To what extent queue properties (i.e. arrival times, service times, and queue discipline) can be discovered from collected data?

- **Hypothesis:** Collecting positional data points using means such as proximity sensor tags attached to field agents (i.e. clients and servers) can reveal arrival times, service times, and queue discipline.
- **Significance:** Formation of waiting lines or queues is inevitable in construction jobsites. One of the most widely used applications of DES is to model waiting lines. Discovering knowledge on the accurate queue properties and their changes is vital in construction DES models.
- **Objective 3:** To develop an activity recognition framework for construction equipment and workers using built-in sensors of mobile phones as a ubiquitous and self-sufficient data collection, storage, and transmission scheme.
 - **Question:** How could activity recognition be executed using existing sensors of ubiquitous mobile phones? In what combination(s) and to what level(s) of detail can activities be recognized and classified using a mobile sensor-based activity recognition technique? To what extent the activity recognition framework is accurate?
 - **Hypothesis:** Recognition and classification of construction equipment and worker activities through built-in accelerometer and gyroscope data is possible using supervised machine learning classification algorithms.
 - **Significance:** An accurate activity recognition framework that takes advantage of readily available sensors in everyday life (i.e. smartphone built-in sensors) that contributes to data-driven simulation input modeling greatly facilitates the implementation of data-driven simulation systems.

1.3 Research Contribution

The main contribution of this research to the body of knowledge is that it lays the foundation to explore the possibility of creating and refining realistic simulation models from complex, unstructured, and evolving operations such as heavy construction and infrastructure projects. The presented framework can be effectively deployed in different stages of a project ranging from planning – by providing a more accurate tool for getting insight into the performance of the system in near future, to operations – by facilitating resource deployment optimization on the fly, site layout configuration and design, and evaluating different operational plans based on the conditions on the ground. The presented research also contributes to the construction simulation research and practice by providing a modeling approach that once accredited by the industry, can serve as a foundation for further work in this area and ultimately, transform human-centered (subjective) decision-making to simulation-based (objective) decision-making. Furthermore, productivity assessment through work sampling, safety and health monitoring using worker ergonomic analysis, and sustainability measurement through equipment activity cycle monitoring to eliminate ineffective and idle times thus reducing greenhouse gas emission (GHG), are some important project performance indicators that can benefit from the developed activity recognition framework.

1.4 Research Methodology

The steps enumerated below outline the methodology pursued in this research to achieve the aforementioned research objectives:

- A preliminary study was conducted to demonstrate the feasibility and applicability of data-driven knowledge-based simulation modeling with focus on data collection using wireless sensor network (WSN) and rule-based taxonomy of activities. The resulting knowledge-based simulation models performed very well in properly predicting key performance measures of real construction systems.
- Time stamped positional data was used to extract knowledge pertinent to client-server interactions in queuing systems and to provide computer interpretable input for corresponding simulation models. Among key properties, queue interarrival times, service times, and discipline (drawing order) were successfully extracted.
- A pervasive mobile data collection technique using built-in sensors of ubiquitous smartphones and data mining was designed, and an activity recognition framework built upon this technique was successfully developed and tested for construction equipment. In this methodology, smartphones were placed inside equipment cabin and accelerometer and gyroscope data were collected while the equipment was working. The collected data were used to train supervised machine learning algorithm and the trained model was applied to unseen examples of data collected from the equipment to accurately predict its state.
- The developed activity recognition methodology was further expanded and revised to design a novel construction worker activity recognition framework. This framework was successfully tested on different categories of worker activities to investigate the accuracy of the framework in detecting complex interactions between field agents.

1.5 Organization of the Dissertation

The following Chapters of this Dissertation are shaped around the concepts, details, and implementation of the research tasks listed above. This Dissertation is divided into seven Chapters. In particular:

- Chapter 1: Introduction – This Chapter contains the background and problem statement, objectives of the research, contributions made by this work to the body of knowledge and practice, and the methodologies pursued in this research to achieve the stated objectives.
- Chapter 2: Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multi-Modal Process Data Mining – This Chapter presents the preliminary study on the feasibility and applicability of data-driven knowledge-based simulation to construction processes. Most of the materials presented in this Chapter are previously published as a technical paper by the author in the American Society of Civil Engineers (ASCE) Journal of Construction Engineering and Management. The paper is reused in this Dissertation with permission from the ASCE (see Appendix A).
- Chapter 3: Evaluation of Queuing Systems for Knowledge-Based Simulation of Construction Processes – This Chapter discusses extracting queue properties for waiting line (queue) formations as one of the most widely used applications of DES in construction projects. Most of the materials presented in this Chapter are previously published as a technical paper by the author in the Elsevier Journal of Automation in Construction. Elsevier does not require any permission to include a previously published paper in a dissertation by the same author.

- Chapter 4: Construction Equipment Activity Recognition for Simulation Input Modeling Using Mobile Sensors and Machine Learning Classifiers – In this Chapter, a comprehensive methodology for employing built-in sensors of ubiquitous mobile phones for data collection and machine learning classifiers for equipment activity recognition and duration extraction is introduced. Most of the materials presented in this Chapter are previously published as a technical paper by the author in the Elsevier Journal of Advanced Engineering Informatics. The paper was in press at the time of writing this dissertation.
- Chapter 5: Smartphone-Based Activity Recognition and Classification of Construction Workers – In this Chapter, construction workers’ activity recognition using mobile phone sensors and machine learning classifiers is investigated. Due to the complexity of workers’ interaction in construction processes, a detailed and thorough evaluation of different classification algorithms is conducted in this Chapter.
- Chapter 6: Data-Driven Simulation of Construction Processes with Complex Worker Interactions Using Smartphone Sensor-Based Activity Recognition – In this Chapter, a relatively complex operation is modeled in a data-driven simulation linked to collected accelerometer and gyroscope data. The added value of using process knowledge obtained from mobile sensory data in simulation modeling is then discussed. For this purpose, the output of the data-driven simulation model is compared to the output of a conventional simulation model that is almost identical to the data-driven model with the exception that activity durations are estimated based upon expert judgements and the dimensions of the workplace.

- Chapter 7: Conclusions – This Chapter concludes the dissertation by providing a summary of the problem statement, key findings, and contributions in previous Chapters.

CHAPTER 2: KNOWLEDGE-BASED SIMULATION MODELING OF CONSTRUCTION FLEET OPERATIONS USING MULTI-MODAL PROCESS DATA MINING¹

2.1 Introduction

Within the construction engineering domain, the use of discrete event simulation (DES) to model resource interactions and operational logic has been the subject of several studies [1, 2, 6]. A DES model is an event-based representation of project activities that constitute an engineering system. Considering factors such as complexity and scale, and given the multidisciplinary nature of a construction or facility project, decision-makers and field engineers may rely on computer-generated simulation results to study key performance indicators including resource allocation, equipment utilization, site planning, and bottleneck elimination [7]. Depending on information availability at different project stages, the level of detail and the scope of resulting simulation models may vary. The granularity and credibility of results generated by these simulation models is a critical factor in determining whether such models can be readily used by project decision-makers [8].

An extensive literature review conducted by the author revealed that most construction simulation models are mainly used during the early planning and design, as they are built upon rigid assumptions and design parameters (e.g. precedence logic, activity durations). For instance, prior to launching a DES model, one has to carefully identify different activities and diligently

¹ *The materials of this Chapter have been previously published in:* [5] R. Akhavian, A.H. Behzadan, Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining, J. Constr. Eng. Manage. 139 (11) (2013) 04013021.

create an activity cycle diagram (ACD) that prescribes the flow of resources. All modifications to this logic as a result of changes in the real system must be manually done which may prove to be a tedious task if not at all impossible. Similarly, activity durations must be provided prior to running the model. Moreover, there is often no systematic way to refine the simulation by overwriting its variables based on the actual conditions on the ground. In all such cases and in the absence of an inclusive methodology to incorporate real field data as the construction evolves, modelers rely on data from previous projects, expert judgments, and subjective assumptions to generate simulations that can reasonably predict future performance. These and similar shortcomings have to a large extent limited the use of traditional DES tools to preliminary studies and long-term planning of construction projects. A longstanding research challenge is how to generate simulation models that are responsive to real time changes in the project during the execution (construction) phase. What makes this fundamental question of utmost importance is that the construction phase of many engineering projects may in one way or the other be affected by uncertainties such as weather delays, safety incidents, unforeseen site conditions, and equipment breakdowns that are not easy to mathematically formulate ahead of time and predict before commencing the actual project. In fact, previous studies have indicated that on average, construction schedules and plans experience changes up to 70% [9]. Therefore, proper simulation-based operations-level planning and control during project execution require that attributes of the corresponding simulation model elements are modified with progress of the project, so that ultimately, the simulation model be completely adaptable and responsive to the latest site conditions. This underlines the importance of a robust methodology that supports the prospect of reliable collection and processing of field data, and effective extraction of relevant

contextual knowledge. Recent advancements in field data acquisition and remote sensing technologies alleviate the challenge (i.e. required time and cost) of manual data collection. For instance, researchers have recently explored different sensor technologies for material tracking [10, 11], human motion tracking [12], labor and equipment tracking [13, 14, 15], and vision-based detection and tracking [16]. However, the majority of previous research has targeted certain project tasks such as controlling delivery and receipt of construction materials, enhancing safety, progress monitoring, and productivity assessment. In light of this, research is still needed to design generic and robust data sensing and handling strategies that yield maximum amount of information from minimum volume of data [17, 18] . Without transforming raw data to process information and ultimately contextual knowledge, collecting large volumes of sensor-based data can merely provide little value for project planning and optimization.

2.2 Main Contributions to the Body of Knowledge

In the past, the problem of transforming raw process data into contextual knowledge, and using the extracted knowledge to automatically generate simulation models has been the subject of some studies within manufacturing and industrial engineering domains [19, 20, 21]. Unlike manufacturing where the production environment is fully controlled and ambient factors are kept to a minimum, construction projects take place in unstructured environments that are hard to comprehend and formulate ahead of time. Thus, simulation modelers often tend to use simplifications, assumptions, and prescriptive parameters to build construction simulation models. Although this approach may streamline the modeling process, it may as well take away from the flexibility and extensibility of the model, negatively impact its accuracy in representing

the project dynamics, and ultimately be detrimental to the model reliability, verification, and validation [22]. Hence, the main contribution of this research to the body of knowledge is that it lays the foundation to systematically overcome these challenges by exploring whether it is possible to create and refine realistic simulation models from complex, unstructured, and evolving operations such as heavy construction and infrastructure projects. This will be achieved by introducing a framework capable of automatically generating and updating simulation models based upon the latest field data collected using a ubiquitous distributed sensor network mounted on construction fleet. These heterogeneous datasets are fused into a reasoning process which extracts contextual knowledge necessary to generate or refine simulation models. The generated simulation model is constantly updated using new incoming data streams. The presented material builds upon author's previous work [23] by (1) establishing a framework for extracting contextual knowledge from raw streaming multi-modal field data, and (2) eliminating the burden of manually creating and continuously updating simulation models by enabling automated generation of realistic models from evolving engineering systems.

2.3 Methodology

In this Section, the key components of the designed methodology are discussed and the current state of knowledge is presented to highlight the main departure points of the presented research.

2.3.1 *Multi-Modal Data Collection and Fusion*

Human brain justification is the major tool and the best example of data fusion in action [24] in traditional simulation paradigms to determine required parameters and variables. Hence, what a

modeler's brain does in defining key parameters of a simulation model is, admittedly, fusion of heterogeneous data from multiple sources including previous project databases, engineering judgment, site layout parameters, and effect of ambient factors. The basic concepts and existing techniques of multi-modal data acquisition and fusion have been investigated in several research studies which aimed at introducing solutions to specific problems within construction engineering. For example, Kannan and Vorster (2000) explored developing an experience database to fuse payload, temperature, and cycle-time data for the load activity in an earthmoving operation. However, haul and return activities were not included in their work as positional data was not collected. Moreover, since data were collected using dump trucks' pre-installed on-board instrumentation (a.k.a. OBI), no additional information describing for example the interaction between dump trucks and loaders, and operational logic were provided. In another study, as-design spatial information were fused with as-is laser scanner spatial data to detect construction defects [25]. Researchers also worked with positional data from global positioning system (GPS) and radio frequency identification (RFID) to estimate the coordinates of construction equipment and inventory items [26]. More recently, spatial (e.g. soil type) and temporal (e.g. weather) data were fused to support construction productivity monitoring [27]. Although all such studies explored data fusion techniques within the construction engineering domain, none investigated a systematic method to collect and synchronize data from multiple resources of different types in order to discover knowledge about the ongoing activities (e.g. state of resources, operational logic) and reveal potential patterns existing in constantly streaming data streams.

The main focus of the research presented here is to demonstrate the suitability and reliability of multi-modal process data collected from active fleet in heavy construction projects (e.g. road construction, pavement resurfacing, earthmoving, mining) in generating and refining realistic DES models describing such operations. In the developed methodology, three modes of heterogeneous process data (i.e. position, orientation, and weight) collected from a distributed network of sensors are synchronized to determine the state of resources (i.e. equipment) that are involved in various stages of an arbitrary operation. The extracted knowledge will be used to generate and update a simulation model corresponding to the real engineering system. As described in the following Sections, what distinguishes the presented data collection and handling framework from existing methods is that in order to initiate the reasoning process, minimum (if any) prior knowledge about the existing site layout, and location and configuration of different resources is required. The developed technique is capable of intelligently observe (sense) the real system and accordingly build (generate) or refine a computer model that best describes the dynamics and evolving nature of the ongoing operations. Once the data collection process is initiated, incoming data streams from individual sensors are captured and analyzed to the extent that a sufficient level of operational knowledge about the ongoing processes can be secured. This initiation stage is an essential component of the framework. The goal is to train the system with minimum amount of incoming raw data necessary to sufficiently describe (with good confidence) the nature and logic of ongoing site activities. As previously discussed, unlike existing methods that use collected data for localization or context awareness, here data is used to provide information and knowledge necessary to generate a realistic simulation model considering that the level of detail and amount of collected data are subject to change as the

project evolves. Hence, computational efficiency and cost are among major factors in selecting the most appropriate data collection strategy that can be sustained for the lifecycle of the project. Very often, captured data is of tremendously large volume and contains high noise ratio [24], such that data cleaning and analysis takes a long time. In the presented research, the goal is to use almost all collected data one way or the other in the reasoning process and consequently, keep the noise and data redundancy to a minimum. Existing videotaping (i.e. vision-based) techniques, as an alternative, may prove to be computationally inefficient for the purpose of this research, as the level of detail and volume of unnecessary collected data (in each video frame) may easily exceed the computational efficiency requirements. For instance, while in the presented work, all collected data is directly relevant to the knowledge extraction process, in vision-based techniques, a large amount of irrelevant data (e.g. background scene) is inevitably collected which may not contribute to the intended purpose of the data collection task. In particular, most computer vision techniques model images as 2D arrays of intensity values (i.e. gray levels from 0 or black to 255 or white) which implies that a 1-second video of 30 frames per second with a fair resolution of 640×480 , will contain 9,216,000 integer numbers (32 bits). Processing this volume of data will require intensive computational effort especially when real time or near real time response is of essence. Also, vision-based techniques such as background subtraction fail to differentiate between object types thus making it difficult to detect target equipment [28]. The developed algorithm in this research is able to identify different states of construction equipment regardless of the pose they hold, visual occlusion, or environmental conditions (e.g. illumination), all of which have been in fact identified as major research areas in vision-based tracking [29, 30, 31]. Hence, the data collection strategy developed in this research

was mainly motivated by the need for a reliable and ubiquitous method easily deployable to collect relevant data in minimum time and with the least computational cost. In the developed methodology, three classes of sensing devices were used: a network of ultra-wideband (UWB) receivers and tags to track resource positions, attitude and heading reference system (AHRS) to track resource articulation, and Zigbee-enabled weight sensors to track the amount of transported material.

2.3.2 Knowledge Discovery and Reasoning Process

The core of the developed framework is the ability of extracting meaningful (contextual) knowledge necessary to automatically generate and refine a simulation model that adequately describes the real system. The need for a solid methodology to extract useful and relevant knowledge from a large amount of collected data has been highlighted in the past by several researchers in construction and civil engineering domains [27, 32, 33]. Efficient knowledge extraction requires that relevant datasets are identified and irrelevant/redundant data is eliminated. Soibelman and Kim (2002) indicated that the ability to conduct valid data analysis and useful knowledge discovery depends on the availability of clean relevant data. Similarly, the importance of quality assessment in data extraction during knowledge discovery is emphasized in the literature [33]. Hence, an efficient knowledge extraction mechanism should (1) employ methods that run on the least possible amount of data, and (2) extract reliable contextual knowledge from relevant datasets.

The first step to extract basic operational knowledge about a resource is to detect the state of that resource. In general, the overall state of a resource can be described using a binary classification

of idle or busy. However, as far as operations-level simulation of construction activities is concerned, this classification is too granular and may cause confusion or misunderstanding. Figure 2.1 shows simple equipment taxonomy in a typical earthmoving operation. As shown in this Figure, knowing that “a truck is idle (not moving)” (in the absence of any other data) one may easily conclude that it is out of service (e.g. has a flat tire) and needs mechanical maintenance, whereas another logical conclusion could be that it is being loaded by a loader and thus, is not moving. Therefore, at the operations-level, this high-level classification should be further broken down into more meaningful subcategories. Figure 2.1 also shows subcategories for the *busy* state where if certain physical motions are observed, the resource state can be categorized as *busy*.

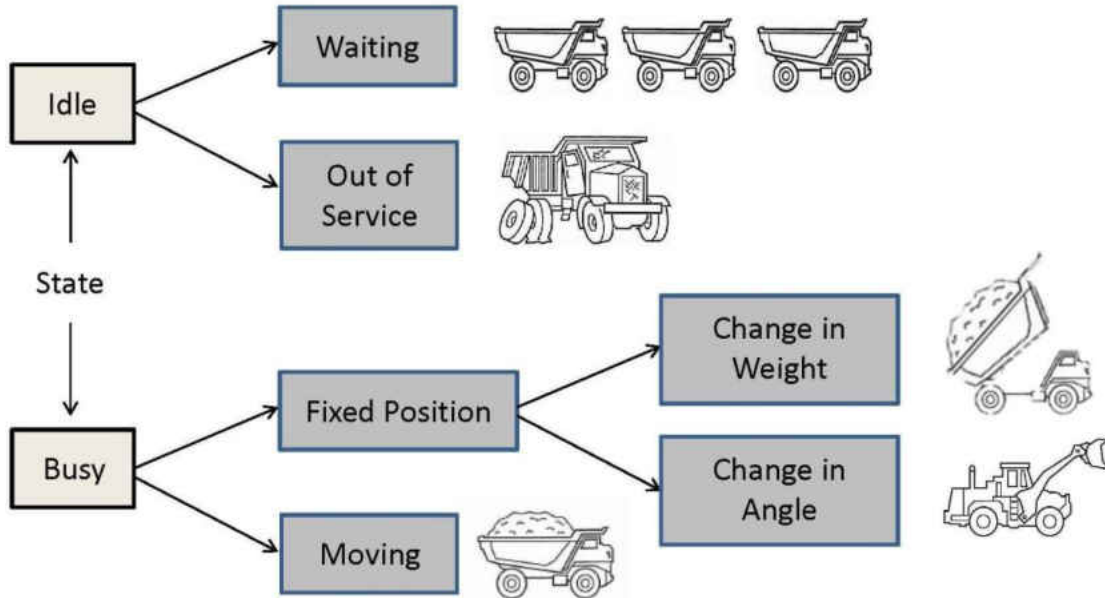


Figure 2.1: Taxonomy-based state classification of construction fleet.

Identifying the correct state of a resource is critical to properly describing ongoing activities in an engineering system. In essence, activities consume resources and their start and end events correspond to when resources are drawn or released by them [34]. The reasoning algorithm observes the trend of data that correspond to each resource state and tries to discover the knowledge (e.g. activity start and end events, duration, resource levels) required to describe project activities. Most often, multiple modes of streaming heterogeneous (i.e. diverse in nature, content, and format) data may need to be evaluated to determine the true state of a resource. Once the start and end events of an activity are determined, activity durations can be calculated by comparing the time stamps of these events. However, since incoming data is often heterogeneous, it has to be first fused and transformed into a common temporal system before any contextual knowledge can be extracted. Another important issue that must be considered is the overall site layout and the approximate locations of where resources are idle (waiting to be drawn by activities), or busy (already drawn by activities). Generally, intensity of fleet position data can assist in this regard. For example, streaming positional data transmitted from a dump truck shows a higher intensity in waiting, loading, or dumping zones (where it is idle the most), compared to hauling routes (where it is moving). Clustering these intensity data helps determine the boundaries of regions that represent waiting queue, dumping or loading areas, and hauling routes. It must be noted that in a dynamic environment such as a highway project, where the location of work zones and routes change, models that rely on a fixed layout [35, 36] may not result in a realistic output. Using an intensity clustering technique, however, changes in the site layout can be constantly monitored to allow simulation model variables (e.g. haul distances, activity durations) to be accordingly updated.

2.3.3 *K-Means Clustering*

Clustering methods are used in knowledge discovery and data mining [37], pattern recognition and pattern classification [38], and machine learning [39]. There are two major types of clustering algorithms: hierarchical, and partitioning methods [40]. In this research, clustering is used to find work zones in a jobsite where equipment spend most of their times (e.g. loading and dumping areas). As a result, hierarchical methods that produce a set of nested clusters similar to a hierarchical tree are not applicable. Partitioning methods, however, divide data points into non-overlapping clusters such that each point belongs to exactly one cluster [41] and therefore are suitable to detect distinct areas with high population of positional data points. Probabilistic clustering, K-medoids, and K-means are the three subsets of partitioning algorithms. The first two methods are computationally complex and are predominantly used in pattern recognition applications [42]. Moreover, in K-medoids, centroids must be exactly at the center of clusters, whereas in K-means they can be anywhere in the sample space (which is a reasonable condition in a cluster of positional data points). Therefore, K-means which is an iterative two-step algorithm [40] was used in this research. As stated in Equation 2.1, the goal of the K-means algorithm is to minimize the sum of squared error for each cluster,

$$J(C) = \sum_{i=1}^k \sum_{x_j \in c_i} (x_j - m_i)^2 \quad (2.1)$$

where x_j represents an individual point within the cluster C_i , m_i is the mean, and the goal is to minimize score function J for all clusters. Initially, given a set of k means $(m_1^{(1)}, m_2^{(1)}, \dots, m_k^{(1)})$ the algorithm partitions n data points into k clusters by assigning each data point to its nearest centroid. As stated in Equation 2.2, x_p belongs to $C_i^{(t)}$ if it is closer to m_i^t than it is to m_j^t . This is

shown in Equation 2.2 where $C_i^{(t)}$ is the cluster i in the t^{th} iteration, x_p is data point x , and m_i^t and m_j^t are centroids of clusters i and j , respectively.

$$C_i^{(t)} = \{x_p : (x_p - m_i^t) \leq (x_p - m_j^t) \forall 1 \leq j \leq k\} \quad (2.2)$$

Then, the new centers are computed to the sample means of their assigned data points, as calculated using Equation 2.3,

$$m_i^{t+1} = \frac{1}{|C_i^{(t)}|} \sum_{x_j \in C_i} x_j \quad (2.3)$$

The iterative process of assigning data points and readjusting means continues until it converges to a steady state and eventually stabilizes.

2.3.4 *Contextual Knowledge Discovery Using K-Means*

Using the K-means clustering method and provided with prior knowledge indicating the expected number of work zones (e.g. loading area, dumping area, resource waiting zones), it is possible to partition multi-modal streaming data based on their intensity. If more than two modes of data are captured, K-means can be also applied on n-dimensional ($n > 2$) vector data to identify k clusters in the n-dimensional space. In the presented methodology, weight data constitutes the third dimension in addition to the XY coordinates of each point and thus, K-means is applied to position-weight data points located inside a 3D XYW space. Therefore, in finding the locations of designated work areas, weight serves as an important attribute of each point and helps in identifying states of equipment that are located in neighboring (and sometimes, spatially close) zones but have different loading conditions (e.g. loading area where weight is increasing vs. loading queue where weight is constant and close to a minimum value).

As a motivating case, consider a simple earthmoving operation where a front-end loader is tasked with loading a dump truck. Therefore, there are 2 locations where the intensity of points is relatively high, and as a result there will be two clusters: loading area and dumping area. Logically, what connect these two clusters are the routes; the haul route connects the loading area to the dumping area, and the return route connects the dumping area back to the loading area. As the XYW dataset is populated using positional and weight data captured from each work cycle, K-means dynamically calculates the approximate centroids of the two clusters. These clusters are then marked as representing the loading area (marked with abrupt weight increase) and dumping area (marked with abrupt weight decrease). Thus, at least two modes of data (i.e. position and weight) are needed to properly identify active loading and dumping zones. Clearly, not all earthmoving operations are as simple as the one described herein and often, multiple dump trucks are employed to haul soil which may cause waiting queues to form near the loading and dumping areas thus, creating other intense clusters. Moreover, there may be rare situations where dump trucks remain idle in an arbitrary location other than the loading area, dumping area, loading queue, or dumping queue. This place can be a service area where equipment receives periodic or random maintenance. Thus, the data reasoning process should be inclusive to cover all special cases that are likely to occur. As such, a robust complex algorithm is needed to extract proper contextual knowledge. Figure 2.2 depicts various combinations of data modes and trends that result in different states of a dump truck in an earthmoving operation.

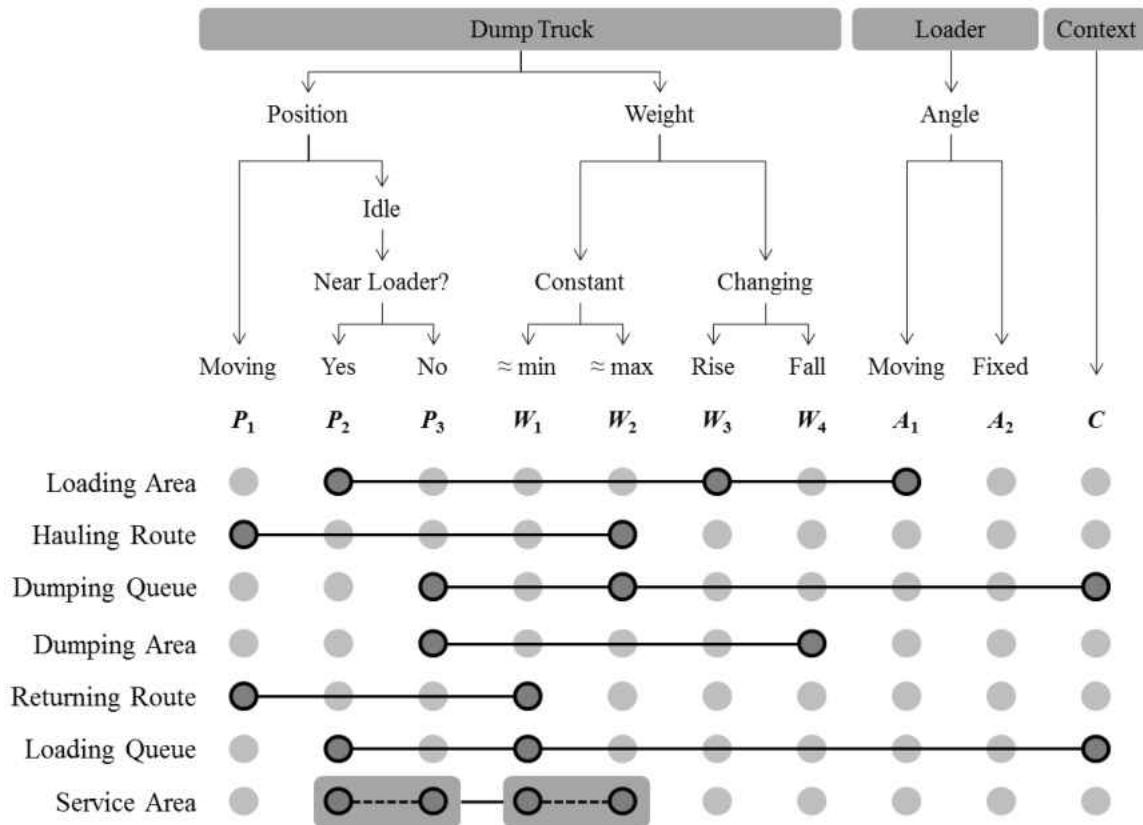


Figure 2.2: Taxonomy of dump truck activities in an earthmoving operation based upon multi-modal process data and operational context.

In this Figure, process data and operational context are linked using solid lines (that represent logical AND) or dashed lines (that represent logical OR). For instance, a dump truck is loaded in the loading area if it is idle in loader’s proximity, its weight is increasing, and when the loader is working (represented by a changing boom angle). This can be represented as $P_2 \cap W_3 \cap A_1$. Likewise, if a dump truck is moving and its weight is close to its maximum value ($P_1 \cap W_2$), it can be concluded that the dump truck is travelling on the hauling route. In dumping area, the dump truck is not moving while its weight is decreasing ($P_2 \cap W_4$). Once the soil is dumped, the dump truck travels back on the returning route which requires the dump truck to be moving

while its weight is close to a minimum value ($P_1 \cap W_1$). This taxonomy provides satisfactory results as far as major activities (load, haul, dump, and return) in an earthmoving cycle are concerned. However, in an operation where multiple pieces of equipment are used, it is very likely that at certain times, dump trucks have to wait in queues before they are drawn to activities. For instance, if a loader is already serving a dump truck, a second dump truck has to wait in a loading queue before the loader becomes available again. Likewise, if the dumping area can only accept one dump truck at a time all other dump trucks that arrive to the vicinity of the dumping area need to first wait in a dumping queue. As stated earlier, another example of a location where dump trucks may remain idle is the service area. If no prior information is provided as to where the service area is located (e.g. somewhere along the hauling route, or along the returning route), then the reasoning process must consider all data and context combinations that may correspond to a dump truck inside the service area. For instance, while one may assume that a stationary dump truck which has a weight close to a maximum value is in a dumping queue waiting to dump its load, in reality, that same dump truck may be idle inside a service area that is located somewhere along the hauling route. To resolve these confusing cases, additional contextual information is needed. A dump truck waits inside a dumping queue only if the dumping area is occupied by other dump truck(s). Hence, what distinguishes dumping queue from a service area is that from the moment a dump truck enters a dumping queue until it leaves the queue, the dumping area is continuously occupied. Using a similar logic, what distinguishes loading queue from a service area is that from the moment a dump truck enters a loading queue until it leaves the queue, the loading area is continuously occupied. All other cases in which a

dump truck is not moving and its weight is constant are considered instances of maintenance or repair work occurring inside the service area, and represented by $(P_2 \cup P_3) \cap (W_1 \cup W_2)$.

Finally, it should be noted that although this reasoning process covers the majority of scenarios involving a dump truck, there are always specific (and rare) cases which may not be detected as expected. However, compared to the majority of activity and queue instances that are correctly identified, the effect of such exceptions on the overall performance of the reasoning process and clustering algorithm is minimal. In any case, if exceptions become rules (i.e. statistically significant), they can be systematically detected and represented as recurring events.

2.3.5 *Automated Simulation Model Generation*

As stated earlier, a major contribution of this research to the body of knowledge is that it will ultimately provide means and methods necessary to conduct (near) real time operations-level planning, look-ahead scheduling, and short-term decision-making by enabling the automated generation of adaptive simulation models using the contextual knowledge extracted from multi-modal datasets. A simulation model generator is a tool for translating real system logic to simulation language, thus enabling computer to represent the behavior of the model [43]. Previous efforts on this topic have been mainly limited to manufacturing and industrial engineering where product trajectories in a structured network of modules were used to generate adaptive manufacturing simulations [19]. In another example, an automated simulation model generator was developed by Son and Wysk [20] for real time shop floor control. Yuan, et al. [21] developed a DES generator for operational systems (SGOS) with applications in manufacturing activities such as fabrication, machine set-up, assembly, and part transportation. Knowledge-

based simulation has been also investigated in creating models for transportation system [44, 45]. Using an input file of natural language (NL) components (e.g. electronics assembly words, expressions, and expectations), Ford and Schroer [46] developed the electronic manufacturing simulation system (EMSS). The earliest use of NL interface, however, was the NL programming for queuing simulations (NLPQ) [47].

Unlike manufacturing and industrial systems where the environment is fully controlled and structured, in many construction projects, resource and operational dynamics, and the presence of ambient factors can intensify uncertainties. Thus, if simulation models are not linked to field data, they will soon become outdated. Despite this, there have been few previous attempts within the construction engineering domain to establish a systematic solution to this problem. For example, a look-ahead scheduling for heavy construction projects was described by Song, et al. [48] in which real time GPS data were used to update a simulation model. In another example, Bayesian updating of input variables of a simulation model was suggested for a tunneling project where the penetration rate of a tunnel boring machine (TBM) was continuously obtained and used to update a distribution function to estimate completion time [49]. However, there has been no systematic research within this domain to evaluate the potential benefits of this subject. In this research, a construction simulation model generator plays a key role as it receives and combines user input (e.g. number and types of resources, project type) and extracted operational knowledge (e.g. activity durations, site layout).

2.4 Results

In order to validate the designed methodology, several laboratory experiments were conducted on a test bed which consisted of a model jobsite and remotely controlled equipment models (see Figure 2.3). Equipment positions were captured by an UWB network, loader boom angle was sensed by an AHRS tracker, and Zigbee-enabled sensors tracked the weight of material transported by dump trucks. Table 2.1 shows basic specifications of these sensors. As listed in Table 2.1, the accuracy of the UWB sensors is 15 cm in a 3D space (and much better in a 2D space). Therefore, considering the update rate of 16 Hz used when conducting experiments in this research, and also given the 12 m² test bed that provided a distance of about 5 meters between loading and dumping areas, the accuracy of the sensor was deemed acceptable. Two such experiments are detailed below and results are provided.

Table 2.1: Specifications of employed sensors in the distributed network.

Sensor	Specifications	
Load Cell	Capacity	5, 10, or 20 Kgs
	Accuracy	± 0.02 %
	Resolution	24 bit
	Update Rate	16 Hz
UWB	Accuracy	15 cm in 3D real time
	Update Rate	0.00225Hz up to 33.75Hz (16 Hz in the experiments)
	Radio Frequencies	Ultra-wideband 6GHz – 8GHz
AHRS	Roll/Pitch Accuracy	0.8° RMS
	Heading Accuracy	0.5° RMS
	Resolution	< 0.5°

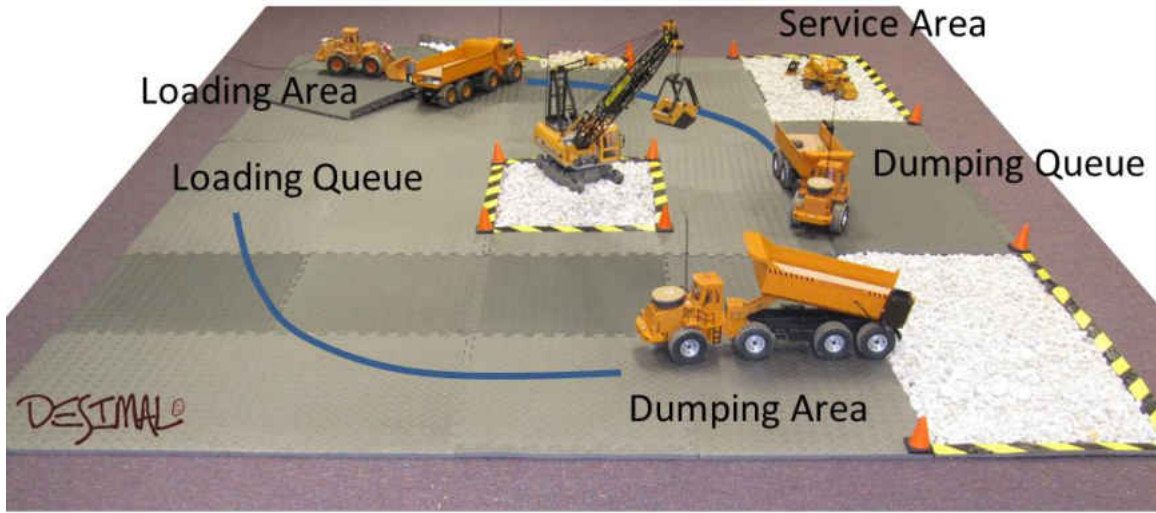


Figure 2.3: Layout of laboratory experiments.

2.4.1 *Experiment 1: One Loader and One Dump Truck, Without Service Area*

In this experiment, the simplest case of an earthmoving operation including one front-end loader and one dump truck was modeled. In each cycle, the loader put soil in the dump truck (inside the loading area), the dump truck hauled the load, dumped it in a designated dumping area, and returned to the loading area to start the next cycle. Hence, there were no waiting queues. It was also assumed that the dump truck would not stop anywhere else on its path due to maintenance-related events. Thus, the K-means algorithm was applied with $k = 2$. In addition, the reasoning algorithm captured the trend of weight data and identified loading and dumping clusters. Figure 2.4 illustrates the plots of collected positional and weight data in 2D (XY) and 3D (XYW) spaces. In the 3D plot, the vertical axis shows weight values. Hence, in the vicinity of loading area, 3D points show a rising trend (increase in weight) while in the vicinity of dumping area, 3D points show a falling trend (decrease in weight). The developed K-means algorithm successfully

detected two clusters (loading and dumping areas) and found the centroids as shown in Table 2.2.

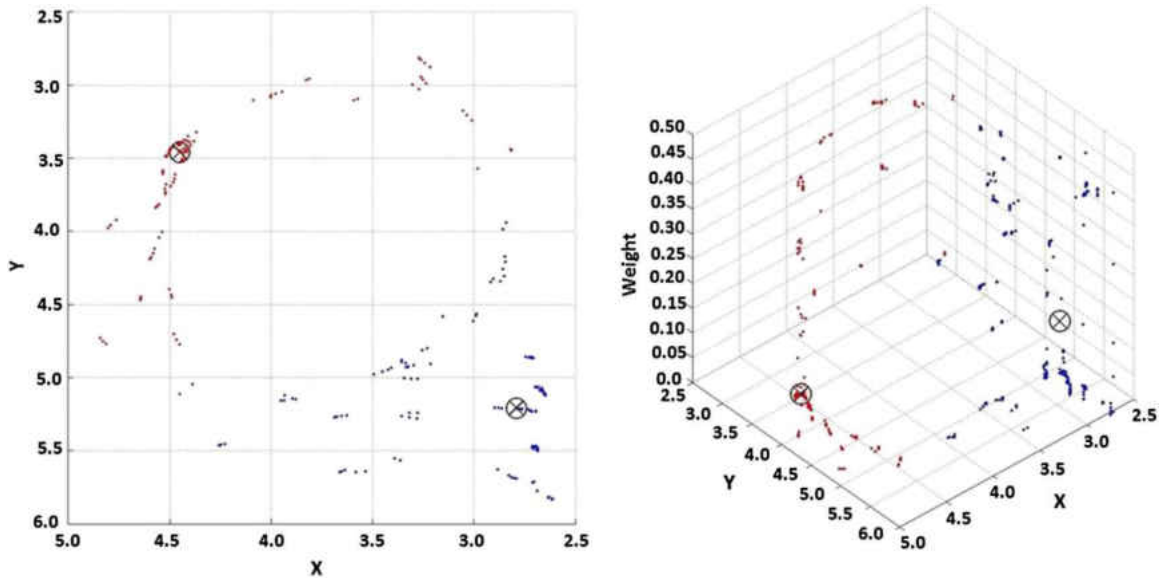


Figure 2.4: Results of K-means clustering algorithm applied to the positional and weight data in 2D (XY) and 3D (XYW) spaces for experiment 1.

Table 2.2: Centroids of the detected clusters using K-means for experiment 1.

Identified Cluster	X (m)	Y (m)	W (g)
Cluster 1	4.45050	3.46007	0.000400
Cluster 2	2.79147	5.20749	0.121125

Considering the load data trend, it was found that clusters 1 and 2 corresponded to the loading area and dumping area, respectively. The developed reasoning technique provided further information about the state of the dump truck which in turn, helped determine activity durations in each cycle. Statistical analysis on pools of activity durations provided mean and standard deviation of each activity duration. Table 2.3 compares observed values (from experiment video)

with extracted values (using the reasoning process) of activity durations. In order to assess if the mean of activity durations obtained from the developed methodology is in good agreement with the observed values, Student's t-test was used [50]. Student's t-test is a popular statistical analysis used to compare means of different populations. In a nutshell, the Student's t-test investigates whether there is a statistically significant difference between the mean values. According to the results, the t-statistic for load, haul, dump, and return activities are 0.32, 0.11, 0.98, and 0.36 respectively and that the p-values of all of them are greater than 0.05 (i.e. confidence level of 95%). This indicates that in the first experiment, the means of observed and extracted durations are not statistically significantly different.

Table 2.3: Observed vs. extracted activity duration means and standard deviations for experiment 1.

Activity	Observed Duration (Seconds)		Extracted Duration (Seconds)	
	Mean	SD	Mean	SD
Load	13.6	8.4	15.4	9.18
Haul	33.8	5.9	34.2	4.92
Dump	11.5	3.4	9.80	1.83
Return	30.4	2.9	32.0	9.49

By capturing sensor data streams and using proper stream mining techniques, the developed methodology can also determine possible changes in the locations of work areas. Road construction is a good example as the cut and fill zones constantly change. Thus, experiment 1 was further evolved by making the dump truck dump soil on different spots. As shown in Figure 2.5, while the location of the loading area was almost unchanged, the centroid of the dumping area moved with time. This change in the layout was captured as more field data was collected.

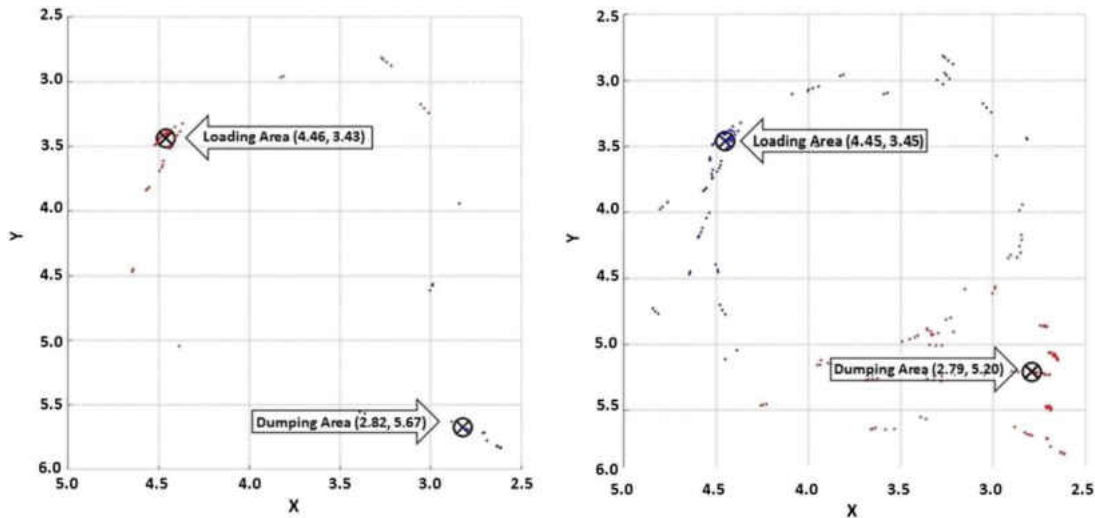


Figure 2.5: Detecting changes in the location of cluster centroids over time.

2.4.2 Experiment 2: One Loader and Multiple Dump Trucks, With Service Area

The second experiment consisted of multiple dump trucks and thus, it was expected that queues would form in the vicinity of loading and dumping areas. In particular, one front-end loader was tasked with loading three (two big and one small) dump trucks, one at a time. A designated service area was also added to test if the developed methodology can properly identify queues from this service area with no prior location information. Hence, it was expected that five clusters representing loading area, dumping area, loading queue, dumping queue, and service area were detected from the streaming fleet data. Therefore, the K-means algorithm was applied assuming $k = 5$. Figure 2.6 illustrates the plots of collected positional and weight data in 2D (XY) and 3D (XYW) spaces. The developed k-means algorithm successfully detected five clusters and found the centroids as shown in Table 2.4.

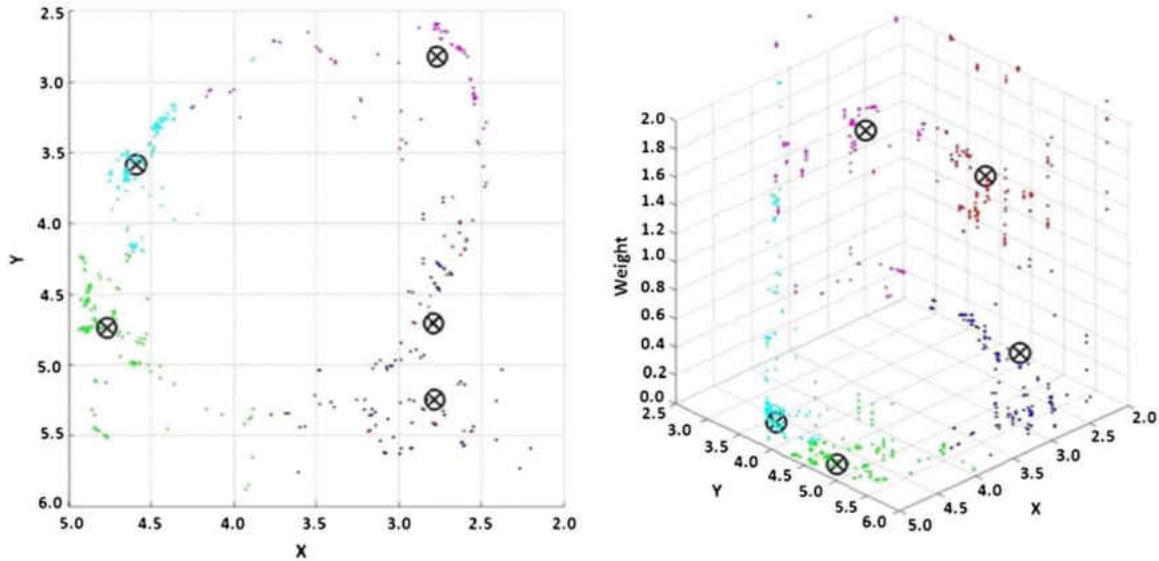


Figure 2.6: Results of K-means clustering algorithm applied to the positional and weight data in 2D (XY) and 3D (XYW) spaces for Experiment 2.

Table 2.4: Centroids of the detected clusters using K-means for experiment 2.

Identified Cluster	X (m)	Y (m)	W (g)
Cluster 1	2.754550	4.42422	1.460313
Cluster 2	4.608870	3.53820	0.000792
Cluster 3	2.929420	5.36794	0.030927
Cluster 4	2.764165	2.77719	1.446068
Cluster 5	4.773650	4.73571	0.001201

Considering the load data trend, it was found that clusters 2 and 3 corresponded to the loading and dumping areas, respectively. Using the state taxonomy shown in Figure 2.2, equipment states were then identified. Next, activity durations were calculated using time-stamped positional and weight data. Statistical analysis on pools of activity durations provided mean and standard deviation of each activity duration. Table 2.5 compares observed values (from experiment video) with extracted values (using the reasoning process) of activity durations.

Using Student's t-test, the t-statistic for load, haul, dump, and return activities are 0.69, 1.10, 1.16, and 1.69 respectively and that the p-values of all of them are greater than 0.05 (i.e. confidence level of 95%). Similar to experiment 1, it can be concluded that there is no statistically significant difference between the two means.

Table 2.5: Observed vs. extracted activity duration means and standard deviations for experiment 2.

Activity	Observed Duration (Seconds)		Extracted Duration (Seconds)	
	Mean	SD	Mean	SD
Load	22.3	10.5	25.7	11.5
Haul	30.9	8.90	34.7	6.33
Dump	10.1	4.80	8.10	2.53
Return	28.2	3.60	35.0	12.2

Time-stamped data contained within the three clusters representing loading and dumping queues, as well as the service area were also used to find the mean and standard deviation of resource waiting times in these locations. Table 2.6 compares observed values (from experiment video) with extracted values (calculated using the reasoning process and statistical analysis) of waiting times. Again, using the Student's t-test, the t-statistics of 1.89, 1.78, and 1.56 are calculated for the waiting times inside the loading queue, dumping queue, and service area. These t-statistic values account for p-values all greater than 0.05 (i.e. confidence level of 95%).

Table 2.6: Observed vs. extracted means and standard deviations of waiting times for experiment 2.

Location	Observed Waiting time (Seconds)		Extracted Waiting Time (Seconds)	
	Mean	SD	Mean	SD
Loading Queue	51.5	15.5	39.4	12.9
Dumping Queue	7.10	1.20	8.30	1.76
Service Area	77.7	4.50	81.0	4.91

2.4.3 *Data-Driven Simulation*

As stated earlier, in order to make a transition from human-centered (subjective) decision-making to simulation-based (objective) decision-making, sufficient amount of project information must be incorporated into simulation modeling. Hence, the next step in validating the results obtained from this research was to evaluate if the outcome can be used to generate more accurate simulation output. This step was essential since a robust and reliable data-driven modeling strategy that can safely replace human assumptions when simulating an engineering system is the first step toward enabling automated generation of DES models. Therefore, results obtained from the first experiment were used to update activity durations inside a DES model that was created in Stroboscope [51]. The ACD of an earthmoving operation is illustrated in Figure 2.7. Two DES input scripts were generated from this ACD. The first script contained activity durations (see Table 2.7) calculated using the actual site layout and resource arrangement (Figure 2.3), distances between work areas, and resource travel speeds. The second script was created using activity durations extracted from construction fleet XYW data and without making any assumptions about the site layout and resource arrangement.

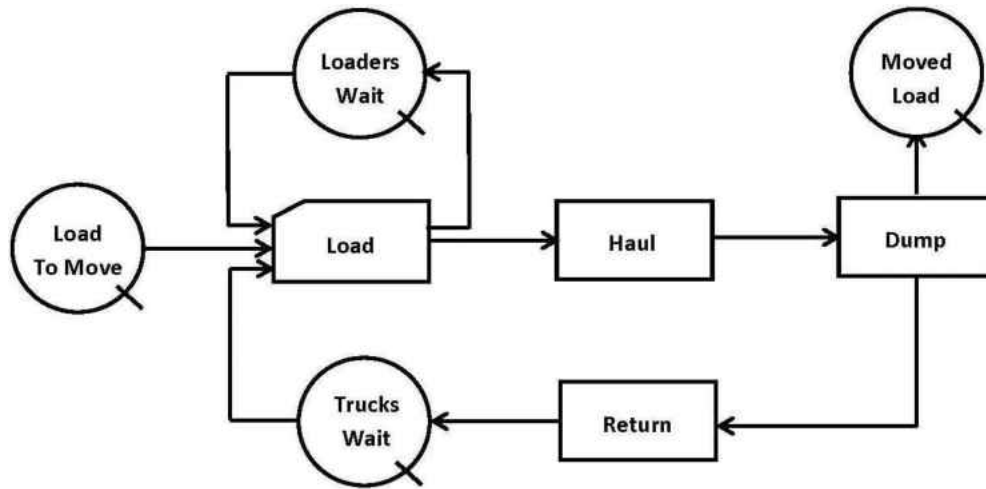


Figure 2.7: ACD of the DES model of a typical earthmoving operation.

Table 2.7: Approximated activity durations based on overall site layout and resource specifications.

Activity	Approximated Duration (seconds)
Load	10.0
Haul	25.0
Dump	5.00
Return	20.0

Also, in both DES scripts, it was assumed that hourly cost of a dump truck and a loader was \$50 and \$135, respectively. As presented in Figure 2.8, four separate measurable quantities (namely total amount of transported soil, loader idle time, total operation time, and total equipment cost) were selected to assess the precision of results generated by the two simulation scripts.

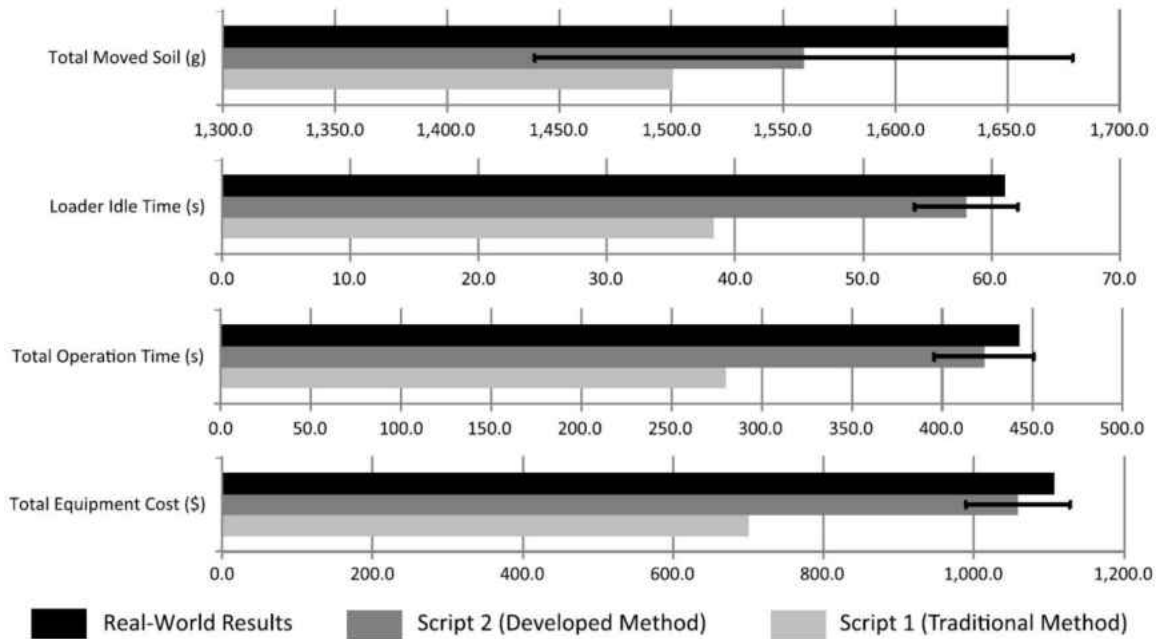


Figure 2.8: Analysis of results obtained from the real-world experiment and the output of traditional and data-driven simulation models.

The error bars for the values obtained from the simulation model based on the second script are also shown in this Figure to indicate the statistical significance of results. As illustrated in Figure 2.8, the output of the simulation model based on the second script (created using extracted duration values) with regards to all four measures was in closer agreement with the observed values from the real experiment. For instance, the total amount of transported soil in the real experiment was observed to be 1,650.0 units (here, grams) in 5 cycles. When this operation was simulated, the output of the first script (created using engineering judgments and resource arrangement) indicated this quantity to be 1,500.5 units, while the output of the second script (created using the developed data collection and mining methods) indicated the same quantity to be 1,559.1 units with a standard deviation of 120.2 units. The same trend was also observed in

the other three measures where the output of the second simulation script was statistically very close or identical to that of the real operation.

2.5 Summary and Conclusions

The convenience of making project decisions based on human expert judgments has caused the construction industry to remain to a large extent reluctant to the prospect of replacing human-centered (subjective) with simulation-based (objective) decision-making. A contributing factor to this problem is that very often simulation models are made when little information is known about a project, rarely updated as the project makes progress, and thus not considered reliable and credible decision-making tools. To overcome these challenges, a systematic approach is needed to enable such models to continuously communicate with the real system, learn from the dynamics of events as they evolve, and accordingly adapt themselves to these changes. To this end, the author investigated the prospect of enabling knowledge-based data-driven simulation model generation and refinement for construction operations. The main contribution of the research presented in this Chapter to the body of knowledge is that it lays the foundation to systematically investigate whether it is possible to robustly discover computer interpretable knowledge patterns from heterogeneous field data in order to create or refine realistic simulation models from complex, unstructured, and evolving operations such as heavy construction and infrastructure projects. The aim of this Chapter was to present a multi-modal (i.e. position, weight, angle) process data mining, fusion, and reasoning algorithm capable of extracting operational knowledge and automatically updating the corresponding DES model. A statistical data point clustering algorithm based on the K-means method was also employed in conjunction

with data mining techniques to discover knowledge about the construction site layout and arrangement of resources. In order to validate the developed methodology, several experiments were conducted. Results indicated that extracted knowledge (e.g. activity durations, resource interactions, site layout) were valid and in good agreement with the reality of the project. Moreover, an earthmoving scenario was modeled in Stroboscope and refined using the discovered operational knowledge, and a comparative analysis was conducted. The analysis revealed that results obtained from the DES script generated using extracted knowledge were more accurate and realistic compared to those from the DES script generated using human-made assumptions.

CHAPTER 3: EVALUATION OF QUEUING SYSTEMS FOR KNOWLEDGE-BASED SIMULATION OF CONSTRUCTION PROCESSES²

3.1 Introduction

Waiting lines or queues exist in almost all industrial and manufacturing processes. In all such queuing systems, there are entities that need to be repetitively processed by other entity(s). The entity waiting in a line to receive service is called a *client* and the entity that processes clients is called a *server*. Similar to queuing systems in manufacturing settings, in many construction systems, clients (or resources) can be delayed in waiting lines when a server (or processor) is already captured by a previously arrived client and thus is busy.

A classic example of a construction queuing system is the arrival of dump trucks in a loading area where excavators or front end loaders load them with soil. As shown in Figure 3.1, cyclic activities of an earthmoving operation consist of load, haul, dump, and return processes. A part of this cycle that embraces the waiting line and server is considered as the queuing system. Therefore, it is clear that the boundaries of the system are not necessarily spatially fixed and can dynamically change depending on the length of the queue and the efficiency of the server.

² *The materials of this Chapter have been previously published in:* [52] R. Akhavian, A.H. Behzadan, Evaluation of queuing systems for knowledge-based simulation of construction processes, *Autom. Constr.* 47 (2014) 37-49.

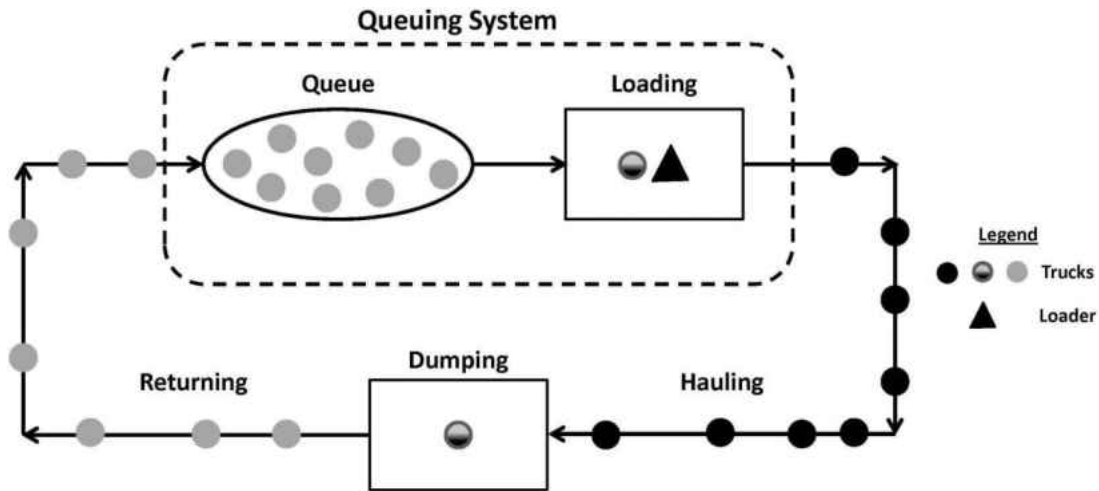


Figure 3.1: Example of a single server queuing system in construction projects.

As soon as a client arrives inside the boundaries of the system, depending on the state of the server (i.e. idle or busy), it either waits in the queue or proceeds to be served immediately. Once the service is completed, the client leaves the system and its state, attributes, and other properties will no longer affect the conditions and properties of the queuing system. That is why in queuing systems terminology, the arrival of a client in the system is also referred to as the client's *birth* and its departure from the system is called the client's *death*, which imply that only the time that a client spends inside the queuing system is of interest to queuing analysis [53]. A final note on Figure 3.1 is that although it shows a construction operation cycle, a queuing system may not be necessarily part of a cyclic operation; that is, the clients that enter the system may not return and the characteristics of the queuing system do not depend on the clients' identifications.

A construction manager who deals with an operation that involves queues is most often interested in knowing the waiting time during which a resource is delayed in a queue, the service time or how long it takes for the server to finish processing a specific client, and the logistics of

the queue (i.e. number of delayed resources in a queue, or the queue length). Such knowledge is of critical importance to allocating the optimal number/type of resources, configuring the site layout, estimating the productivity, and determining the durations of individual operations as well as the entire project. Towards this goal, simulation models have been widely used in modeling queuing systems and to obtain valuable insight into the characteristics of the queues and their impacts on the overall project [54]. As previously stated, this is mainly due to the fact that the processing of clients by a server is a repetitive task and simulation models are perfect tools to predict the performance measures of repetitive processes of undeterministic nature. Among others, discrete event simulation (DES) models are particularly employed in construction and infrastructure projects since most often, the entire construction system can be broken down into discrete processes [6, 55, 56]. Within the construction and infrastructure domain, proper modeling of queuing systems is not a trivial task due to the stochastic, uncertain, and transient nature of such operations. A good example of such stochasticness that happens frequently and needs to be analyzed in the context of construction management is rework [57, 58]. In queuing systems, in order to model the uncertainties in customer arrival times, most mathematical queuing theories suggest the use of specific probability distributions such as the exponential distribution [53, 54, 59]. However, previous research in construction systems based on real world observations of resource arrivals (e.g. dump trucks waiting in line to receive service from a front end loader or an excavator) indicated that the assumption of exponentially distributed arrival times can be often invalid [59, 60, 61, 62]. Moreover, there are other important properties of a queuing system such as the queue discipline that must be accurately modeled when simulating queue operations. In particular, the sequence of queue operations not only can follow any of the

well-known disciplines (which will be described in details in this Chapter), but also can be adjusted occasionally due to spatio-temporal requirements of the jobsite and the real vs. planned work progress [51, 60]. Halpin and Riggs [60] indicate “breaks in queue discipline” as the first challenge among several difficulties in field applications of queuing models. According to Martinez [51], discipline expression in modeling construction queues can be very dynamic and dependent on resource dynamic properties. Therefore, modeling of queuing systems requires accurate input with regard to queue properties that may change over the course of a construction project.

The necessity of providing a simulation model with accurate input data describing queuing systems and client-server interactions under dynamic and uncertain conditions highlights the importance of utilizing adaptive DES models that can be updated and fine-tuned in accordance to operations-level changes occurring in the real system. This requires meticulous data collection and mining processes to enable extracting relevant knowledge necessary to build the simulation model. To this end, this Chapter describes algorithms designed to extract knowledge pertinent to client-server interactions in queuing systems and to provide computer interpretable input for corresponding simulation models. First, a description of relevant previous studies is provided and identified gaps resulting in the presented research is discussed. Next, major properties that characterize a queuing system are introduced and their significance in designing construction and infrastructure simulation models are explained. Then, the algorithms that were designed and implemented to find and represent queue properties inside simulation models are described and the underlying mathematical background is briefly explained. Finally, the robustness and

effectiveness of these algorithms will be examined using empirical data and results will be discussed.

3.2 Research Background

Although utilizing operations-level simulation models that help achieve high levels of efficiency in managing construction projects have been explored and widely advocated in academic research [55, 60, 63, 64], there is still much room for investigating their real value and potential applications that can result in their systematic accreditation by the construction industry [55, 65]. Recent studies tried to investigate the reasons behind the limited and often, isolated use of simulation models in large scale by the industry. Among others, it was stated that most existing construction simulation systems rely on historical data and expert opinions to create simulation models [66]. Given the dynamics involved in most construction systems, such input data may turn out to be unrealistic (resulting in optimistic or pessimistic output), and is often hard to be independently verified. Therefore, the output of the resulting simulation models can be far from the realities of the operations on the ground. In the absence of methods that facilitate the process of constantly updating these simulation models with factual data from the real construction system, such models will soon be obsolete and of little (if any) value to the decision-making process [55, 56, 66]. In order to alleviate this problem, it has been previously discussed that collecting factual data as the project makes progress, discovering meaningful knowledge from this data, and feeding the extracted knowledge to corresponding simulation models can be a promising approach [5]. In order to achieve this, the possibility of collecting, fusing, and mining process data has been recently investigated by the author through developing an integrated

framework for construction equipment data-driven simulation models [5, 67, 68]. There have also been other sparse studies aimed at addressing this problem in limited scopes [69, 70]. Despite these efforts, in almost all previous studies, project resources and entities were considered as single units for data collection and little knowledge was produced from the collected data to describe how individual entities would interact with one another at the process-level over time. AbouRizk, et al. [65] indicated that the first requirement of developing a construction simulation model is acquiring knowledge about the logic and sequence of the operation. Knowing the interactions, relationships, and interdependencies between different entities is a crucial step in acquiring knowledge about the logic and sequence of activities, and can reveal potential predominant work patterns dictated by some entities. Therefore, in the context of queuing systems where entities are in constant interaction with one another, acquiring accurate data to generate knowledge pertaining to the client-server relationships is necessary for developing valid simulation models.

A number of researchers studied the implementation of queuing systems in construction simulation modeling. For instance, in one study, the FLEET program, queuing theory, and DES were used for selection of loader-truck fleets in infrastructure projects [61]. Using DES models, Ioannou [71] investigated the formation of queues during the process of rip-rap placement for the construction of a dam embankment. In another study, the probability distributions of mixer trucks' arrival and service times in concrete delivery and placement were examined and best fit probability distributions for activity durations were identified [72]. In addition, queuing input data uncertainty in earthwork projects was investigated using a probabilistic queuing model with fuzzy input and fuzzy probabilities and also a purely fuzzy queuing model [73]. However, none

of the aforementioned studies explored the potential of providing a DES model with factual data describing queuing systems in order to generate a more realistic simulation model. In the next Section, major queue properties as relevant to the goals and discussions presented in this Chapter are presented.

3.3 Queue Properties

Queues are characterized by a number of properties that represent the interrelationships between the entities involved in a queuing system. The arrival process, service duration, and queue discipline are among the most important properties of a queue [74]. In addition to these basic properties, the numbers of servers, capacity of the queue, and the population of entities to be served are some of the other properties of a queuing system. However, since information related to these latter properties are often provided as part of the project specifications (e.g. site layout and temporary route arrangement may dictate the number of dump trucks that can form a queue close to the loading area at any given time) or equipment manufacturers' catalogues (e.g. bucket capacity of an excavator can be used to determine how many dump trucks can be served within a certain time period), further onsite data collection and analysis regarding these properties do not contribute much to simulation model input data generation and thus are not the main focus of this study.

As previously discussed, queue properties and the ability to detect and use their exact and correct values are essential in designing complex DES models. For instance, *Nonstationary queues* –in which the complications are due to the interactions that occur while the equipment are moving in traffic– in essence are subject to varying measures. In a construction of dam embankment with

nonstationary queues, Ioannou [71] modeled a sophisticated operation using Stroboscope in which activity durations and queue discipline depended on the operation's progress and certain measures over the course of the project. Stroboscope is a programmable and extensible simulation system designed for modeling complex construction operations in detail and for the development of special-purpose simulation tools [51].

3.3.1 *The arrival process*

In a queuing system, the arrival process can be specified by a sequence of interarrival times that are independent and identically distributed (IID) as a simplifying assumption in order to fit probability distributions with fixed parameters. The randomness involved in the arrival occurrences makes it easy for the interarrival times to be characterized by a probability distribution [75]. For convenience, random arrivals are often modeled as a Poisson process with exponentially distributed interarrival times with a rate of λ (number of arrivals in unit time) and mean of $1/\lambda$ (average interarrival time) [53]. Despite its ease of use and widespread application in queue modeling, the exponential distribution may fail to fully reflect the real world observations taken from actual clients' arrivals and interarrival times especially in systems with transient and constantly changing states (e.g. construction projects) [59, 62].

3.3.2 *The service process*

A service facility (i.e. where a server processes clients) can have different number of channels and phases. In the presence of more than one server, channels refer to available routes clients can take after having waited in line to reach the service facility [54]. An earthmoving operation with two excavators is an example of multi-server queuing system that has two channels. Phases are

the number of stops a client must make after getting in line and before the service is completed. For instance, a stockpile of precast concrete segments may need to be moved by a tower crane to another place where a heavy lifter puts them on flatbed trailers. The service facility depicted in Figure 3.1 is single-channel single-phase. In addition, the time it takes for clients to be served by server(s) (a.k.a service time) is another determining factor in a queuing systems. Similar to the interarrival times, service times are IID and can be represented by a probability distribution [75].

3.3.3 *The queue discipline*

Queue discipline is defined as a rule or set of rules based on a specific attribute of the entities. It determines the pattern (i.e. order) by which entities in the queue receive service [74]. The choice of the queue discipline and the rules to be applied can significantly affect the number of entities waiting in a queue, the average waiting time, and the efficiency of the service facility [76]. The most common queue discipline is first-in-first-out or FIFO in which clients in line are served based on their chronological order of arrival. Although FIFO has been long used as a default queue discipline in modeling queuing systems [74], in many scenarios, it is equally likely that clients be served according to other service patterns. For instance, the last-in-first-out or LIFO discipline may be the case in situations where a heap or stack of clients (e.g. raw materials, prefabricated concrete segments, steel sections) are waiting to be processed by a server. Other than FIFO and LIFO, the serving pattern of a queuing system can be characterized according to an intrinsic attribute of the entities in the system. This type of queue discipline is called *priority queues* or PRI queue discipline. In the example of a queue of dump trucks waiting to be loaded by an excavator, priority might be given to those dump trucks with less fuel left. Sometimes, there may be no rule according to which clients receive service from the server, in which case the

queue discipline is considered as service-in-random-order or SIRO. Figure 3.2 illustrates the concepts of FIFO, LIFO, and PRI queue disciplines. In this Figure, clients are specified by letter C and the server is specified by letter S. Each of the three queues shows the client that should be drawn from the queue under the specified queue discipline.

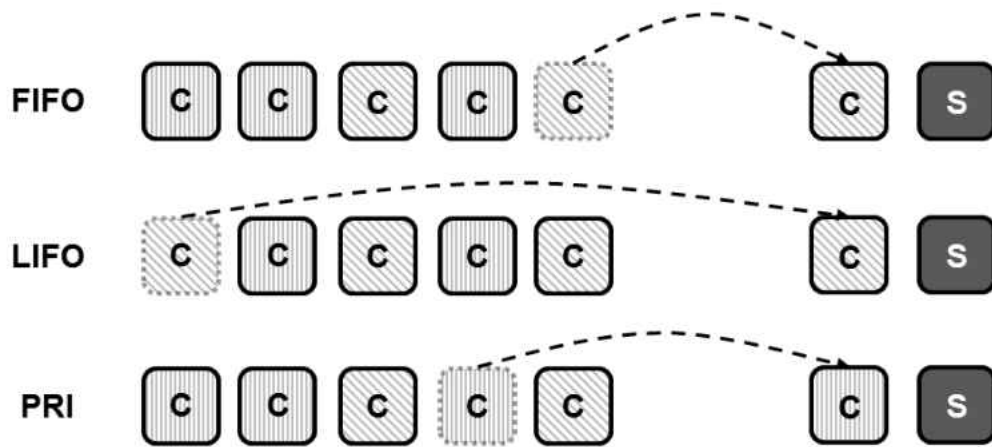


Figure 3.2: Demonstration of FIFO, LIFO, and PRI queue disciplines.

Also, Figure 3.3 summarizes the properties of the queuing system shown in Figure 3.1 in each stage of the process.

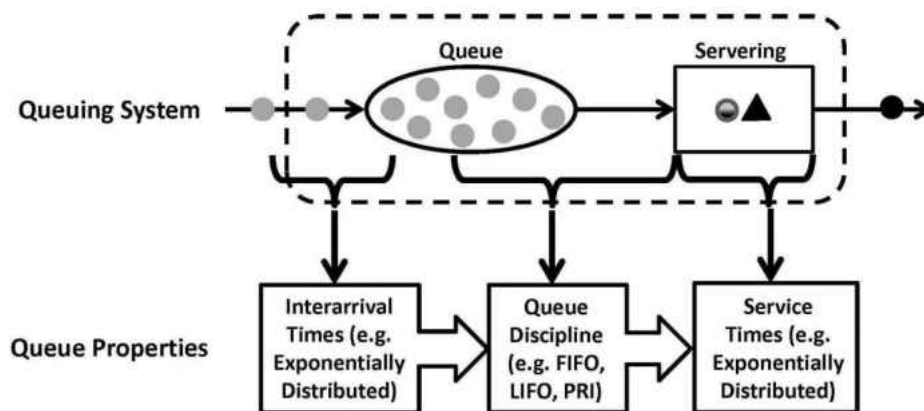


Figure 3.3: Sequence and the properties of a queuing system.

Considering the diversity of queues, a notational system is widely used to distinguish between different systems. An abridged version of the notation is based on the $A/B/c/D$ format, in which A represents the interarrival probability distribution, B represents the service time probability distribution, c represents the number of parallel servers, and D represents the queue discipline [77]. For consistency, this notational system is used throughout this Chapter. Inside a simulation model, interarrival times and service times may be represented by probability distributions and queue discipline can be defined using specific functions provided by the DES platform. In the next Section, the process of finding the best mathematical representations of interarrival times and service times will be discussed.

3.4 Mathematical Representation of Interarrival and Service Times

In data driven DES modeling, data necessary to describe the arrival process and service durations of the queuing system should be collected from the real world system. For the purpose of this research, details of data collection and knowledge extraction methodologies designed and already validated by the author are described in [5]. Law and Kelton [62] indicated that in case of having access to actual data on a certain input random variable, three approaches can be adopted to specify a distribution corresponding to the available data and use the distribution in a DES model accordingly. The first approach is to use one of the observed data values every time it is needed in the simulation. The second approach is defining an empirical distribution function based on the collected data and sample from that distribution whenever that input data is required in the model. Finally, the third approach is fitting a standard theoretical distribution to the collected data points. Generally, all the aforementioned methods have their uses in different

applications and none can be ignored. However, in the context of DES modeling of construction and infrastructure systems, and considering the stochastic and uncertain nature of activities, the first approach will almost certainly result in a biased model with unrealistic representation of interarrival times or service durations. The second approach may seem to be viable for adoption at the first glance. However, there are shortcomings associated with this methodology that makes it unsuitable for the purpose of this study. In the context of this work, in particular, since an empirical distribution function is just representing the values that exist in the pool of collected data, the quality of the distribution relies solely on the quality of the sample. As a result, there might be some irregularities in the distribution function. This causes even more problems when the number of collected data points is small. Moreover, an empirical distribution function can only generate values inside the range of the collected data. For example, there might be a few instances of very large service times that may have occurred before or after the data collection period, and thus will not be reflected in the empirical distribution function [62]. This is not desirable in designing a simulation model because the performance of a model depends significantly on the potential of predicting an extreme event that may not be part of the empirical data but the probability of which can still be captured at the tails of a theoretical probability distributions [62]. When a theoretical distribution is used, if it is extremely unlikely for a variable to exceed a certain value, say y , the distribution can be truncated at y to better represent the reality. Therefore, it is imperative that fitting a standard theoretical distribution to the collected data can better guarantee a more realistic sampling of the observed values inside a simulation model.

Collected data points and their corresponding probabilities can be shown by means of histograms [62]. In some cases, it is possible to find one or more probability distributions that best match a histogram. However, when dealing with more than one candidate distributions that look representative, a closer evaluation of the level of fitness is necessary for accurate input modeling. A commonly used approach to systematically assess the quality of a fit is using goodness-of-fit tests. A goodness-of-fit test evaluates a null hypothesis, H_0 , specifying that the random variable conforms to the assumed distribution function against the H_1 hypothesis that states the opposite [8, 74]. The procedure consists of calculating a test statistics and comparing it with a critical value to see if it exceeds that value and thus, there is sufficient evidence to reject the null hypothesis.

There are three commonly used goodness-of-fit tests for evaluating the quality of fitness; 1) Chi-Square test, 2) Kolmogorov-Smirnov (K-S) test, 3) and Anderson-Darling (A-D) test. The goodness of this comparison is appraised based on the distribution of test statistic as shown in Equation 3.1, which approaches the chi-square (χ_0^2) distribution with k-1 degrees of freedom [74].

$$\chi_0^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i} \quad (3.1)$$

This test is based on the null hypothesis of no considerable difference between the expected and sample distributions. The universal applicability of the Chi-Square test makes it a commonly used method, however, since the test depends on the number of intervals, it is usually advised to use this test in addition to other tests and with special care [8, 62, 74].

The Kolmogorov–Smirnov (K-S) test compares the experimental cumulative distribution function (CDF) with the CDF of an expected theoretical distribution [74]. The empirical CDF $S_N(x)$ is defined by Equation 3.2,

$$S_N(x) = \frac{\text{number of random } R_1, R_2, \dots, R_N \leq x}{N} \quad (3.2)$$

The K-S test is based on the largest absolute deviation (D) between $F(x)$, the CDF of the theoretical distribution and $S_N(x)$, that is based on test statistic [74] as shown in Equation 3.3,

$$D = \max |F(x) - S_N(x)| \quad (3.3)$$

Both K-S and Anderson–Darling (A-D) tests directly compare the hypothesized model’s distribution to the empirical distribution; however A-D places more emphasis on the differences found in the tails. The test statistics for A-D is defined by Equation 3.4,

$$A^2 = -n - \frac{1}{n} \sum_{i=1}^n (2i - 1) [\ln(F_x(x_i)) + \ln(1 - F_x(x_{n+1-i}))] \quad (3.4)$$

in which, x is the random variable, n is the number of the data points, and A^2 is the test statistics. It is generally accepted that the A-D test provides better results than K-S and Chi-Square tests [8, 62]. All three goodness-of-fit tests have been previously applied to construction simulation problems [72, 78, 79]. However, despite its better performance, the A-D test was not extensively used in construction simulation modeling [80]. In this study, all three tests are employed to find the best fit probability distributions.

3.5 Queue Property Discovery Algorithms

3.5.1 *Interarrival and service times*

In this research, data that are collected using localization multimodal data collection sensors include, among others, the coordinates of each tracked entity (i.e. clients and servers) identified by its unique ID number at any point in time. This data collection and storage technique creates a pool of unique data points that contains ID numbers attributed by coordinates $x_t^{ID}, y_t^{ID}, z_t^{ID}$ that indicate the longitude, latitude, and altitude of each entity (the current discussion only concerns the position of entities in 2D, but the third dimension is also collected and can be useful in some cases). As soon as an entity enters the boundaries of the queuing system, its arrival time, t_a , is marked and stored. Depending on the availability of the server, it then either enters the queue or proceeds directly to the server. More detailed discussion about the reasoning process deployed to discover the exact state of an entity in the system and all modes of data contributing to the knowledge discovery phase can be found in [5]. In a nutshell, the reasoning process is a taxonomy-based approach that first finds the state of construction resources, and then, according to the stream of multi-modal data extracts the contextual process knowledge. If the entity joins the queue, its waiting time is calculated by subtracting the time it starts receiving service, t_s , from t_a . If no queue was formed and the server is free waiting for the next client, there is no waiting time associated with this client in the queuing system and thus $t_s - t_a = 0$. Depending on the service duration, the client leaves the server at t_f when its processing by the server is finished, and the duration of the service can be calculated as $t_f - t_s$. Therefore, the timing status of the client in the system can be represented as $ID (t_a, t_s, t_f)$. When the service starts, the

coordinates of the client should be close to those of the server. This condition confirms that the client is in fact in the process of receiving service at that moment. For example, for the client 020 (09: 46: 16, 09: 47: 03, 09: 48: 35) which is served by server ID 1, the following conditions must be met at the time of service,

$$x_{t_s}^{20} \cong x_{t_s}^1 \quad , \quad y_{t_s}^{20} \cong y_{t_s}^1 \quad (3.5)$$

Considering such spatio-temporal relationships between clients and server(s), the waiting time and service time can be obtained. The interarrival time of the queuing system can be also calculated by taking into account the arrival pattern of all clients over a specific period of time (i.e. duration of real system observation). In particular, the difference between the arrival times of any two subsequent clients can be stored as a new interarrival time value.

3.5.2 *Queue disciplines*

As discussed earlier, there are a variety of rules that determine the order according to which clients are drawn from a queue for further processing by the server. Compared to the FIFO and LIFO situations that are the two most common queue disciplines, PRI queues introduce more complexity in terms of the interaction between clients and server(s) and thus, require more investigations [81, 82]. In this Section, the methodology designed for discovering all such queue disciplines from the incoming client-server data streams is described.

As previously stated, clients' arrival times (t_a) and service start times (t_s) can be extracted by capturing positional data. Having obtained data from several service instances, the chronological record of collected t_a and t_s values is used to create and populate two separate lists hereafter

referred to as the *Order of Arrivals (OA)* and the *Order of Services (OS)*. These two lists serve as the backbone of all algorithms described in this Section. In essence, knowledge revealing a specific queue discipline is discovered through a side-by-side analysis and comparison of entries and their chronological orders in these two lists. The general algorithm for discovering the true queue discipline is represented in the flowchart shown in Figure 3.4.

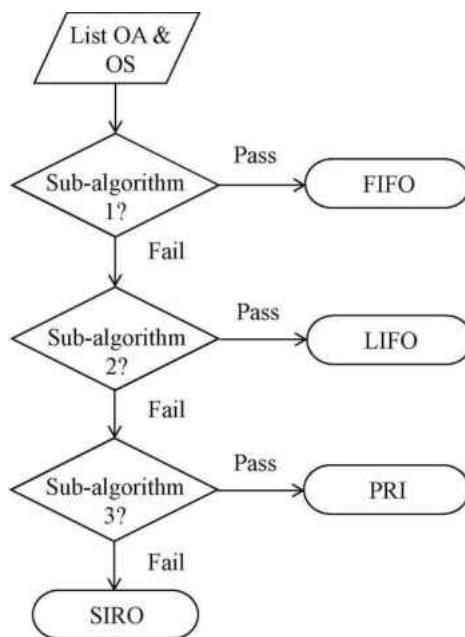


Figure 3.4: Main flowchart for queue discipline discovery.

Following the flowchart of Figure 3.4 and given an *OA* list, each of the three sub-algorithms (corresponding to FIFO, LIFO, or PRI disciplines) first creates an *Expected Order of Service (EOS)* list. The generated *EOS* list is then compared to the actual *OS* list (as generated from the real data). Depending on the degree of compatibility (i.e. similarity) between the *EOS* and *OS* lists, each sub-algorithm outputs an accuracy level by which the collected data from the real system resembles the queue discipline represented by that sub-algorithm.

3.5.2.1 First-in-first-out (FIFO) queues

FIFO is often considered as the default queue discipline in most simulation software [74]. Discovering a FIFO discipline by analyzing the *OA* and *OS* lists is relatively simple, as illustrated in the flowchart of Figure 3.5.

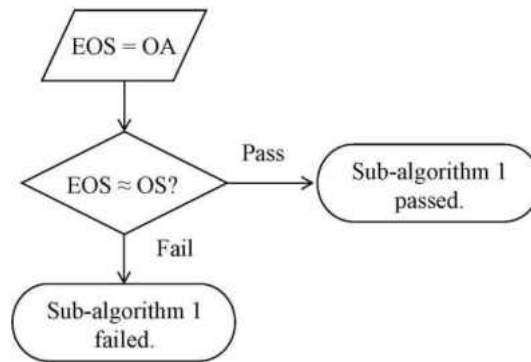


Figure 3.5: FIFO queue discipline discovery flowchart.

According to this Figure, the *EOS* list in this case should be identical to the *OA* list as the clients receive service in the same order they arrive in the queue. Therefore, the algorithm verifies if the actual *OS* list is compatible with the *EOS* list. If each and every element (i.e. client IDs) in the *EOS* list is equal to the corresponding element in the *OS* list, FIFO is the queue discipline (with 100% accuracy level) that best describes the client-server interactions represented by the collected data.

3.5.2.2 Last-in-first-out (LIFO) queues

For a queue with LIFO discipline, drawing a conclusion is more involved than the previous case. The rule states that the entity that comes last should be served first. The main complexity in this case lies in the timing of server becoming available (i.e. giving service to the last client has just

ended) and the timing of clients arriving in the queue, that may result in scenarios in which the *EOS* list is almost never generated by inverting the *OA* list. In particular and given a queue formation with n clients and a free server, the server must process the last client (i.e. client n) first. While processing client n , if a new client arrives in the queue (with $n-1$ clients left) then the next client to be served is always the very last client in the queue (i.e. new client n). However, if no new client arrives in the queue between two instances of server availability, then the next client to be served will be the second to last client (i.e. client $n-1$) since the very last client (i.e. client n) has just been served by the server. In general, if no new client arrives in the queue between k instances of server availability, then the next client to be served will be the k th last client ($k-1$ clients have already been served by the server).

Figure 3.6 shows the flowchart of the LIFO discipline sub-algorithm. As shown in this Figure, once the next available service time is read from the *OS* list, the client that arrives last in the queue is spotted. If the ID of this client is not among those who have been already served, it will be added to the *EOS* list as the next client to be served. Otherwise, it is concluded that no new arrivals has been occurred in between two consecutive server availability, and thus the algorithm looks for the client who arrived one before last. This iteration continues until there is no more service time available and all clients have been already processed by the server.

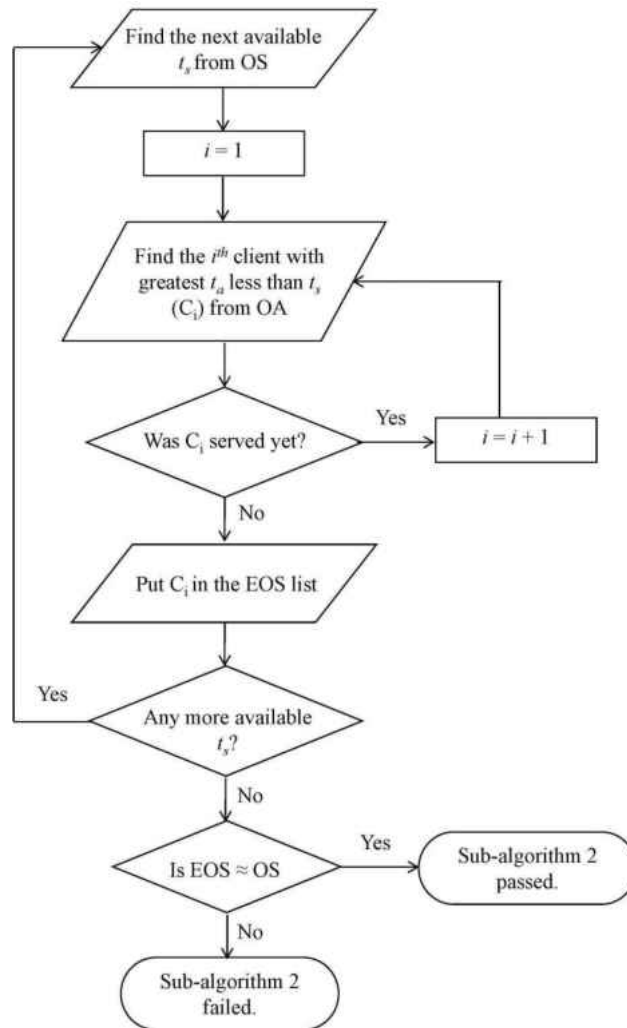


Figure 3.6: LIFO queue discipline discovery flowchart.

3.5.2.3 Priority (PRI) queues

Priority queues are slightly different from queues with FIFO and LIFO disciplines since an external rule determines the order by which clients receive service. The external rule is often defined such that it assigns higher priorities to some clients over others based upon an attribute intrinsic to clients (e.g. size, type) or combinations of attributes [82]. A FIFO or LIFO queue is in fact a special case of a PRI queue in that the attribute used to prioritize the clients is the time

of arrival in the queue. In its simplest form, an algorithmic approach for discovering the PRI discipline in a queuing system starts with considering two channels for the clients that arrive in the queue. In practice, this resembles the case where clients with higher service priority are channeled through a designated line while all other clients are lined up in a regular line. A common example of such channels is the passenger boarding process at an airport or bus terminal, where passengers whose boarding passes show “premier access” can board at any time of their choice using a special lane [83]. In this research, and in order to design a generic approach to discover queue discipline in PRI queues, the notion of two physically-separated channels is, however considered secondary. In other words, it is not necessary for clients to form two separate lines; the sub-algorithm developed for this queue discipline can properly function even if all clients are placed in a single line and in any arbitrary order after entering the boundaries of the queuing system.

Figure 3.7 illustrates a priority queuing system in the moment the server has just become available (i.e. the client in service has just left the boundaries of the queuing system). Here, clients that are sorted in the ellipse based on their t_a (smaller t_a is placed closer to the server) are of two types: square and circle. Let’s assume that squares have higher priority over the circles. Therefore, they can be grouped into two imaginary lines: a priority line and a regular line. Let’s also assume that each of these two imaginary lines follow a FIFO discipline, internally. Essentially, before any client is drawn from the regular line for processing, the designed algorithm constantly checks whether there is any client waiting in the priority line. Therefore, in the queuing system of Figure 3.7, assuming that during the next four server availability instances, no new client is added to the queue, all existing clients can be enumerated according to

the order by which they will receive service. Clients of type square receive lower service numbers (higher priority) and clients of type circle receive higher service numbers (lower priority).

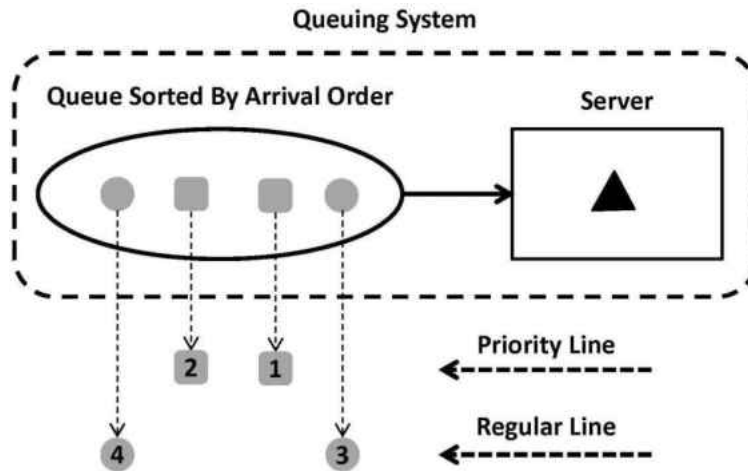


Figure 3.7: Assigning priorities to clients in a PRI queuing system.

Figure 3.8 shows the flowchart of the PRI discipline sub-algorithm. It should be noted that based on the general flowchart presented in Figure 3.4, generated *OA* and *OS* lists are processed by all three sub-algorithms for all queue disciplines. As previously stated, in each case, accuracy level will be calculated for every queue discipline (i.e. FIFO, LIFO, and PRI) based on the degree of compatibility between the *OS* and *EOS* lists. The queue discipline with the highest accuracy level can then be used as the dominant queue discipline when simulating the real system. Consequently, if none of the queue disciplines has a high enough accuracy level, it can be inferred that no specific rule was used to draw clients for processing. In this case, the SIRO discipline can be used to describe the queuing mechanism in the corresponding simulation model. In the following Section, a comprehensive queuing scenario will be presented to better

describe how raw collected data is transformed and used by the queue discipline discovery algorithm, and how accuracy levels are calculated for each potential queue discipline.

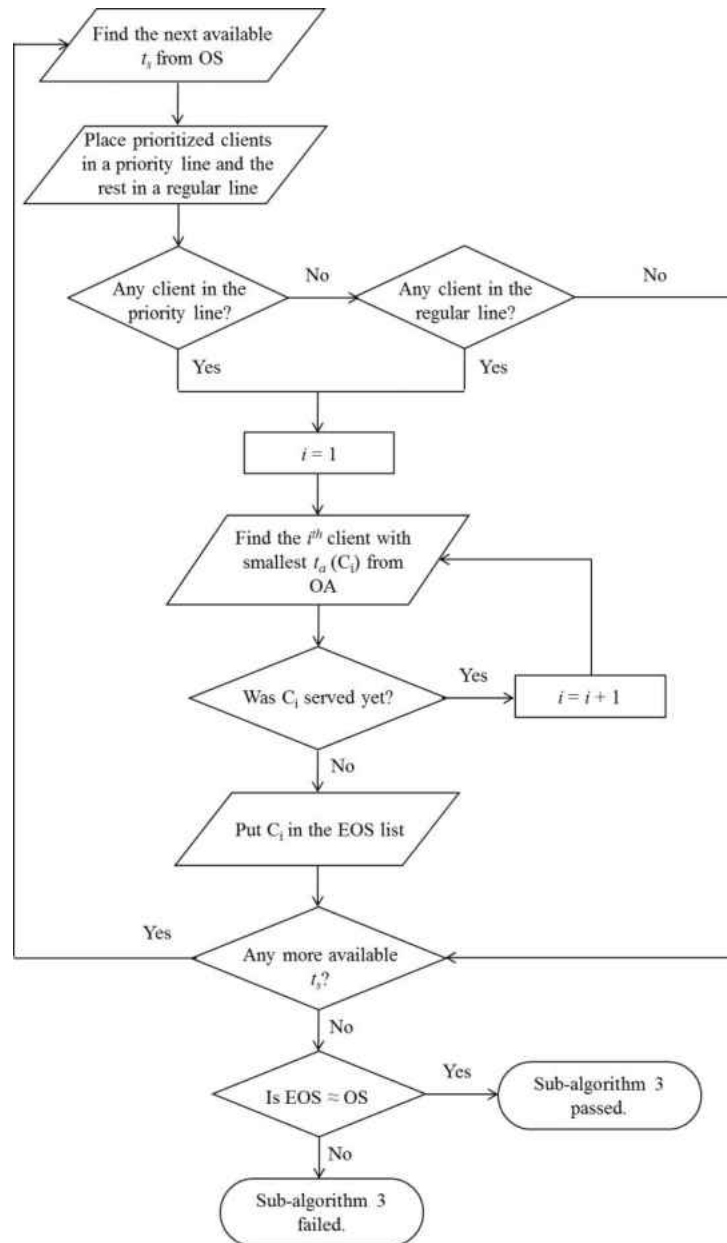


Figure 3.8: PRI queue discipline discovery flowchart.

3.6 Empirical Analysis and Validation of Results

In order to evaluate the functionality of the designed queue property discovery algorithms in extracting key queue parameters (i.e. interarrival times, service times, and queue discipline), a series of experiments were conducted in the Decision Support, Information Modeling, and Automation Laboratory (DESIMAL) at the University of Central Florida. In each experiment, client-server interaction data was collected from the real system under a prescribed queue discipline, and the performance of the designed algorithms in correctly identifying the queue discipline as well as other queue parameters from the incoming data streams were examined.

Several experiments were conducted using commercially available ultra-wideband (UWB) receivers and tags continuously moving in a 3D space and representing clients arriving at, waiting in, and leaving a queuing system, as well as server(s) in charge of processing clients. The location of each tag was recorded in real time with an accuracy of 15 cm in 3D (XYZ) space. The update rate of the sensors was set to 16 Hz and a total of 11 tags were employed in each experiments. Out of the 11 tags, 10 were used to model clients and 1 was used as a server. In total, more than 78,000 time-stamped positional data points were collected.

Table 3.1 shows a sample chronological record of the data points collected in a two-second time interval in one of the experiments. The first column in this Table shows the current time in hh:mm:ss format, the second column is the ID of the UWB tag, the third, fourth, and fifth columns contain the 3D coordinates of the tag, and the last column is the tag status. As soon as a tag entered the boundaries of the queuing system, its status was marked as “Just Born” (e.g. tag

148-011 in Table 3.1). Otherwise, if it already existed in the queuing system, its status was shown as “In System”.

Table 3.1: Sample of the collected data points.

Time	Tag ID	X	Y	Z	Tag Status
18:28:06	154-195	3.91023	3.04932	0.15026	In System
18:28:06	147-220	3.85169	3.04145	0.149724	In System
18:28:06	147-212	3.08748	3.02152	0.178556	In System
18:28:07	147-252	3.62637	3.0435	0.265829	In System
18:28:07	147-020	3.89411	3.06027	0.132566	In System
18:28:07	147-207	3.87354	3.06197	0.148414	In System
18:28:07	147-175	3.79864	3.08372	0.13733	In System
18:28:07	147-211	3.706	3.06462	0.118912	In System
18:28:07	154-195	3.90496	3.04561	0.1512	In System
18:28:07	148-011	4.65458	3.95913	0.316904	Just Born
18:28:07	147-220	3.8502	3.04176	0.155713	In System
18:28:07	147-212	3.08696	3.02207	0.179014	In System
18:28:07	147-252	3.62177	3.04439	0.265192	In System
18:28:07	147-020	3.89573	3.06664	0.131962	In System
18:28:07	147-207	3.87055	3.06544	0.151006	In System
18:28:07	147-175	3.80224	3.08201	0.13583	In System
18:28:07	147-211	3.70344	3.06286	0.120768	In System
18:28:07	154-195	3.90306	3.04175	0.156936	In System

Figure 3.9 shows a schematic illustration of the experiments’ test bed. In this test bed, the queue area that hosts the 5 cm by 5 cm UWB tags (specified by C letters) is separated by 15 cm from the service area that where another UWB tag acts as a server (specified by S letter).

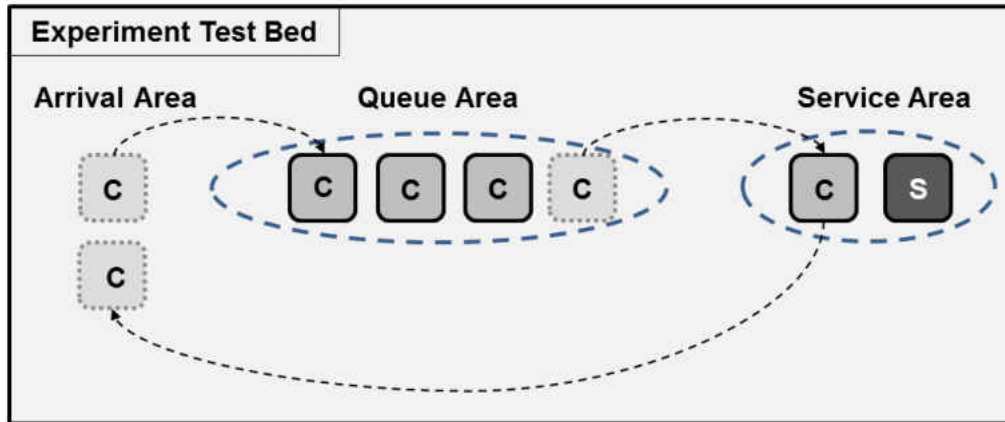


Figure 3.9: Schematic illustration of the experiment test bed.

In each scenario, a series of 50-60 instances of arrival (birth), service, and departure (death) was examined by human experiment assistants. Interarrival and service times were determined and scaled by observing the same number of instances in a real world earthmoving operation in which dump trucks waiting in a queue received service from a single excavator. More specifically, an assistant who was tasked with organizing the arrivals of entities used a stopwatch to add each entity to the queue area according to the scaled interarrival times observed in the real world earthmoving operation. Another assistant was responsible for drawing entities from the queue and putting them adjacent to the server tag. Again, the timing of this task was according to the scaled service durations taken from the real world earthmoving operation. Subsequent time-stamped arrivals of the client tags to the Queue Area determines the interarrival times and the time during which a client tag is in the Service Area (closest to the server) determines the service duration. Movement of tags does not affect the sensing accuracy.

Initially, three distinct experiments were conducted with FIFO, LIFO, and PRI queues, respectively. In each case, order of arrivals and services, and interarrival and service times were

extracted using the methodology and algorithms described earlier. Moreover, the queue discipline in each case was discovered with a satisfactory accuracy level. Furthermore, once the performance of the algorithms in extracting the true queue discipline was verified, a fourth experiment was conducted in which the queue discipline was changed in the middle of the experiment to evaluate the performance of the designed algorithms in detecting the change as it occurred in the real system. A good example of why this experiment is relevant to scenarios from the real world is the case where a project manager, due to a variety of reasons (e.g. saving fuel, minimizing engine wear), decides to change a regular FIFO queue discipline to a PRI discipline that is based on the *fuel left* attribute of the dump trucks waiting to be loaded by an excavator [51]. Knowing the exact point in time when this queue discipline change occurs in the real system can help a modeler update the corresponding simulation model accordingly to better reflect the actual conditions on the ground, thus resulting in more accurate simulation output and more realistic performance prediction. In the designed experiment, the implication of the priority queue was such that UWB tags with even tag IDs were prioritized over the ones with odd tag IDs.

Table 3.2 shows a sample of the *OA* and *OS* lists along with the corresponding interarrival and service times extracted in one of the initial three experiments. Note that this Table only contains a small portion (three-minute time interval) of a much larger record of data points collected during the experiment.

Table 3.2: Sample of the OA, OS, interarrival, and service times extracted in Experiment 1.

Arrival Order	Arrival Times	Interarrival Times	Service Order	Service Start	Service Finish	Service Duration
147-020	18:25:38	0:00:17	147-020	18:27:27	18:27:39	0:00:12
154-195	18:25:55	0:00:21	154-195	18:27:42	18:27:47	0:00:05
148-011	18:26:16	0:00:19	148-011	18:27:50	18:27:54	0:00:04
147-212	18:26:35	0:00:07	147-212	18:27:57	18:28:17	0:00:20
147-252	18:26:42	0:00:07	147-252	18:28:20	18:28:26	0:00:06
147-211	18:26:49	0:00:32	147-211	18:28:30	18:28:37	0:00:07
147-175	18:27:21	0:00:08	147-175	18:29:01	18:29:11	0:00:10
147-220	18:27:29	0:00:07	147-220	18:29:14	18:29:27	0:00:13
147-207	18:27:36	0:00:15	147-207	18:29:30	18:29:48	0:00:18
147-020	18:27:51	0:00:05	147-020	18:29:52	18:30:00	0:00:08
154-195	18:27:56	0:00:11	154-195	18:30:03	18:30:17	0:00:14
148-011	18:28:07	0:00:28	148-011	18:30:21	18:30:35	0:00:14
147-212	18:28:35	0:00:12	147-212	18:30:39	18:30:42	0:00:03

In each experiment, up to six commonly used standard theoretical probability distributions (i.e. Beta, Erlang, Exponential, Gamma, Normal, and Triangular) were fitted to the interarrival and service times. In each case, the quality of the fit was evaluated using the three goodness-of-fit tests (previously described) in the @RISK probability distribution fitting software. Next, probability distributions were ranked among each other according to their test statistics. Finally, out of the six representative probability distributions, the one that received the highest average rank was selected to be used inside the simulation model to describe the undeterministic characteristics of the queuing system. For example, Table 3.3 shows the results obtained for the first of the initial three experiments.

Table 3.3: Ranking of the Best fitted probability distributions for Experiment 1.

	Test	Beta	Erlang	Exponential	Gamma	Normal	Triangular
Interarrival Time	<i>Chi-Square</i>	5	1	2	4	6	3
	<i>K-S</i>	1	3	4	2	5	6
	<i>A-D</i>	2	3	4	1	5	6
Rank		8	7	10	7	16	15
Service Time	<i>Chi-Square</i>	3	N/A	4	N/A	1	2
	<i>K-S</i>	3	N/A	4	N/A	1	2
	<i>A-D</i>	2	N/A	4	N/A	1	3
Rank		8	N/A	12	N/A	3	7

As shown in Table 3.3, for the first experiment, both the Gamma and Erlang distributions received the highest rank among all distributions that were fitted to interarrival times and thus, either of the two was deemed suitable for use in the simulation model. Also, for service times, the Normal distribution had the best relative ranking among others. Moreover, the test statistics for both the Gamma/Erlang and Normal distributions were smaller than the critical value at 95% accuracy level in all three tests which implies that there is no reason to reject the H_0 hypothesis (i.e. there is no reason to reject the assumed distribution). In Table 3.3, the term N/A indicates that for that particular distribution, the data convergence failed and thus the distribution could not be fitted to the data. Figure 3.10 shows the Gamma and Normal distributions as fitted to the extracted interarrival and service times in experiment 1, respectively. It should be noted that, although Normal distribution was selected as the best fit distribution, when using it inside a simulation model, appropriate lower and upper bounds should be defined, so not to sample extremely low or large values. This can be done by using a truncated distribution with the same

main parameters or adding control statements in the simulation script to ignore sample values smaller or larger than user- specified bounds.

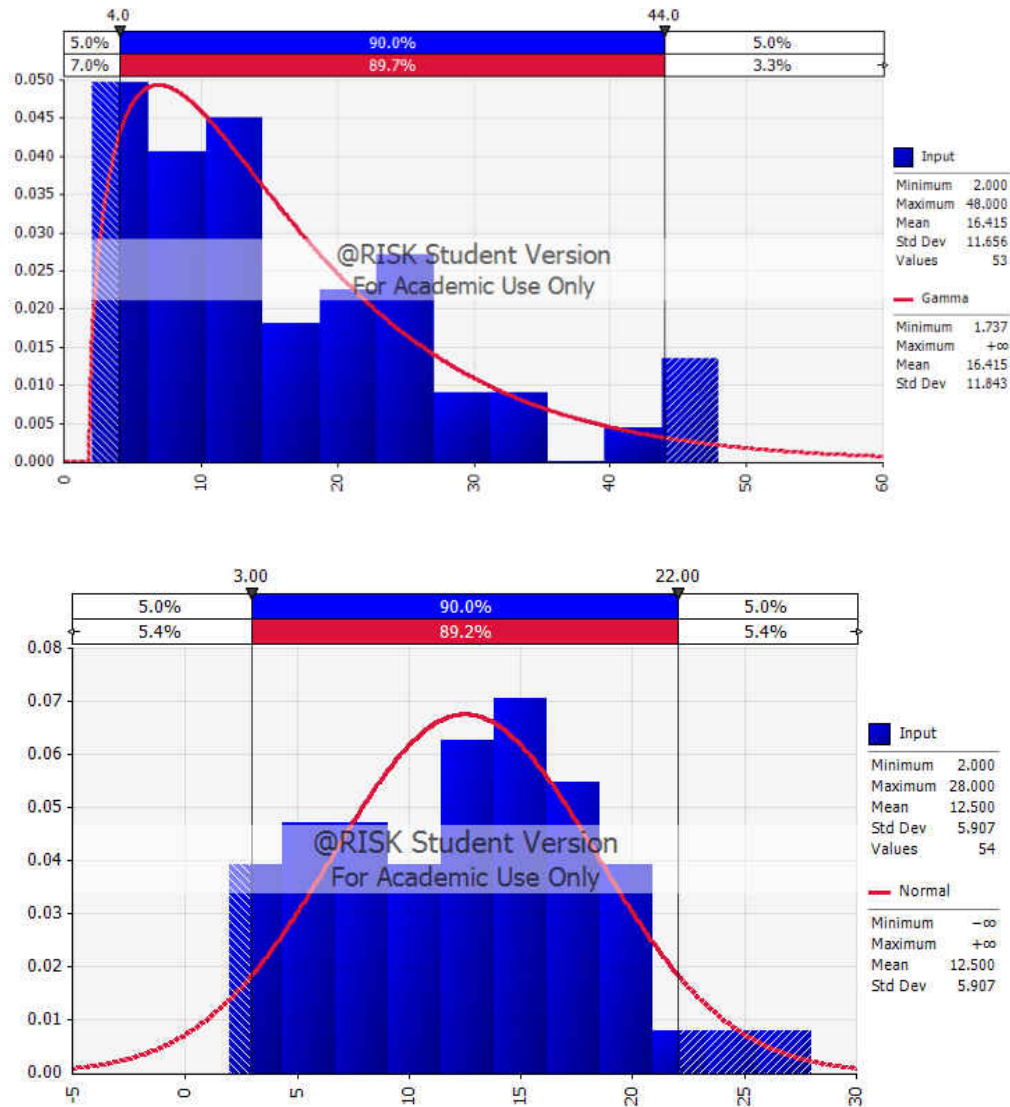


Figure 3.10: Gamma Distribution Fitted to the Extracted Interarrival Times in Experiment 1.

The same procedure was followed for the next two experiments. Table 3.4 lists the best fitted distribution for the interarrival and service times for all initial three experiments. Note that all of the selected distributions as summarized in Table 3.4 had a test statistics less than the critical

value in all three goodness-of-fit tests which implies that in none of them, the H_0 hypothesis can be rejected. In addition, in order to validate the selected probability distributions, the cumulative distribution function (CDF) of the actual empirical distribution constructed from the experimental data was compared against the final distribution selected using the two-sample K-S test statistic. Results are shown in the fourth column of Table 3.4, and the CDF graphs for the interarrival and service times of the first experiment are illustrated in Figure 3.11.

Table 3.4: Best fit distribution for Experiments 1, 2, and 3.

Experiment	Queue Parameter	Best Fitted Distribution	Two-Sample K-S Test Statistic
1	Interarrival Times	Gamma [1.5,9.5]	0.09
	Service Times	Normal [12.5, 5.9]	0.06
2	Interarrival Times	Gamma [1.7, 8.1]	0.07
	Service Times	Beta [8.0, 3.5]	0.06
3	Interarrival Times	Erlang [2, 6.8]	0.16
	Service Times	Normal [16.16, 6.2]	0.11

The two-sample K-S test used to compare the empirical data and the selected distribution should not be confused with the on-sample K-S test in Table 3.3 that was used as a goodness-of-fit measure. The critical value of the two-sample K-S test for the level of significance of $\alpha = 0.05$ is given by Equation 3.6, in which D_α is the critical value at $\alpha = 0.05$, and n_1 and n_2 are the sizes of the samples [84]. Here $n_1 \approx n_2 = 55$ and thus $D_\alpha \approx 0.26$. According to Table 3.4, all the test statistics are less than this critical value and thus the null hypothesis (samples are from the same distribution) is not rejected.

$$D_\alpha = 1.36 \sqrt{\frac{n_1+n_2}{n_1 n_2}} \quad (3.6)$$

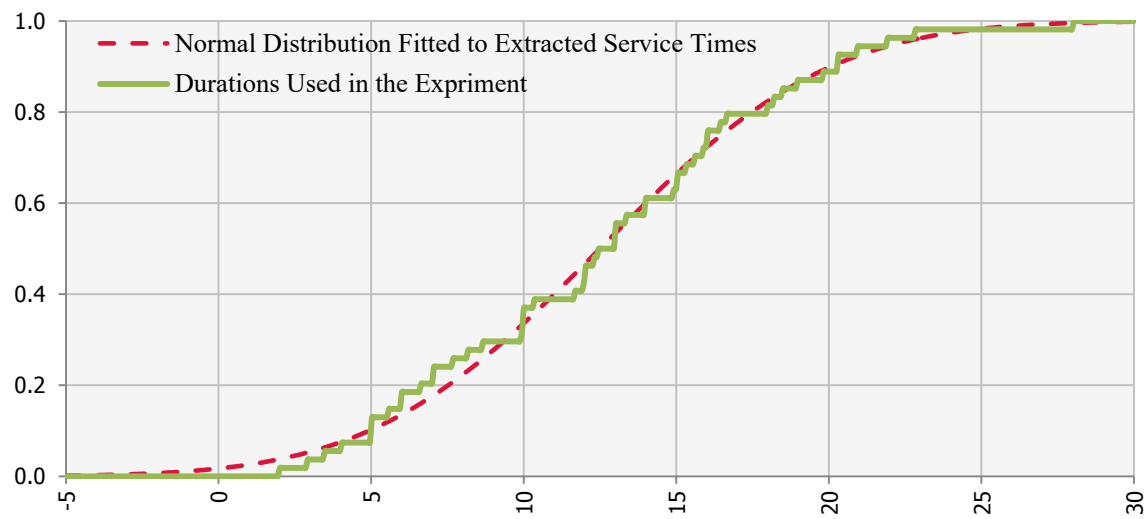
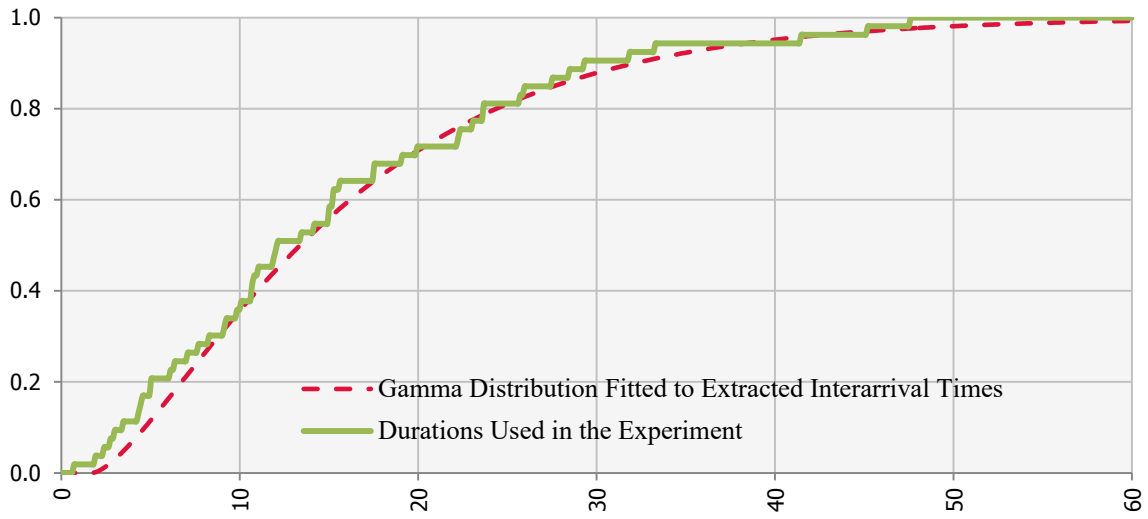


Figure 3.11: CDF of the experiment input data and the best fit distribution to the extracted durations.

Once probability distributions were fitted to the interarrival and service times, the robustness of the designed methodology in discovering the true queue discipline was evaluated. As mentioned earlier, in the initial three experiments, FIFO, LIFO, and PRI queue disciplines were prescribed to the real system. Once time-stamped client-server positional data was collected, the *OA* and *OS* lists were generated and inputted to the queue discipline discovery algorithm illustrated in Figure 3.4. Results indicated that the queue discipline of the first experiment (which was set to be FIFO in the real system) was discovered to be FIFO with 100% accuracy level. In essence, when *OA* and *OS* lists of this experiment were processed by sub-algorithm 1 in Figure 3.4, it was observed that in 100% of the times the queue discipline was in fact FIFO. Also, the accuracy levels calculated by sub-algorithms 2 and 3 for the same *OA* and *OS* lists of experiment 1 were 5.60% and 8.30%, respectively. The results obtained from the queue discipline discovery algorithm for all initial three experiments are summarized in Table 3.5.

Table 3.5: Results of the queue discipline discovery.

Actual Queue Discipline (Real World)	Calculated Confidence Level for the Discovered Discipline			Normalized Confidence Interval
	<i>Sub-algorithm 1 (FIFO)</i>	<i>Sub-algorithm 2 (LIFO)</i>	<i>Sub-algorithm 3 (PRI)</i>	
FIFO	100%	7.54%	8.37%	86.32%
LIFO	16.66%	94.80%	2.21%	83.39%
PRI	16.98%	9.43%	97.23%	78.64%

As shown in Table 3.5, all three sub-algorithms were successful in detecting the correct queue discipline that was adopted in each experiment. Also, in each case they detected the other two queue disciplines with relatively very low accuracy levels. Therefore, in order to draw a fair

conclusion about the performance of the algorithms in each experiment, the last column represents the normalized accuracy levels considering all the results obtained in all sub-algorithms.

Finally, in the last (i.e. fourth) experiment, a change in the queue discipline was implemented at an arbitrary point in time during the experiment. In particular, while the queuing system was originally set to function under the FIFO rule, the discipline was switched to PRI when 50% of the service instances were fulfilled. Under the new queue discipline, the UWB tags with even tag IDs received higher priority over those with odd tag IDs. It was detected that the interarrival times and service times followed Gamma distribution and Normal distribution, respectively. Also, results of the data mining process indicated that a FIFO discipline was predominant in 54.7% of service instances and a PRI discipline was detected in 61.3% of service instances. Through normalizing the results, the output of the algorithm revealed that in the first 47.15% of service instances, a FIFO discipline was followed (compared to 50% as observed in the real experiment) while in the remaining 52.85% of service instances, the dominant queue discipline was PRI.

Results obtained from experiment 4 were imported into the Stroboscope simulation system. Table 3.6 shows the input parameters of the simulation model based on the extracted interarrival and service times as well as the dynamic queue discipline, as extracted from the data collected from the client-server interactions in the real system.

Table 3.6: Input parameters of the simulation model for Experiment 4.

Queue Parameter	Simulation Input
Interarrival times distribution	1.475 + Gamma (1.75, 8.10)
Service times distribution	Normal (13.66, 8.02)
Queue discipline	FIFO (47.15%) + PRI (52.85%)

The queuing system was modeled based on the activity cycle diagram (ACD) shown in Figure 3.12. The Gamma distribution for the interarrival times was used to model the duration of the `Arrival` activity, and the Normal Distribution was used to model the duration of the `Service` activity. Also, the queue discipline was used to draw clients from the `ClientsWait` queue. Figure 3.13 shows the Stroboscope simulation script in which the implementation of the discovered queue discipline change in the model is highlighted (shown in front of the `DISCIPLINE` statement).

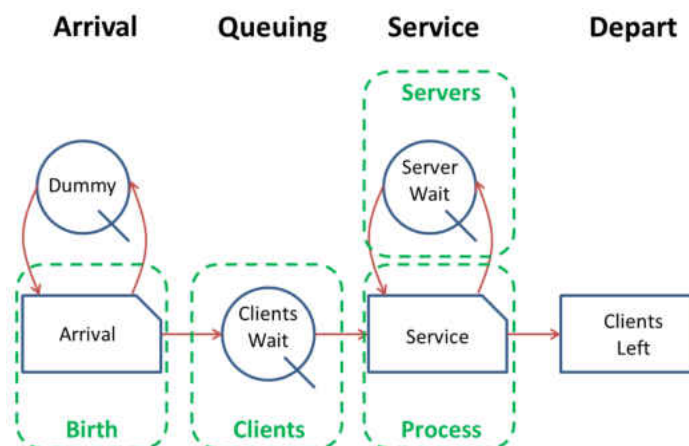


Figure 3.12: Activity cycle diagram of the queuing system in Stroboscope.

```

/* Durations for the FIFO-PRI Test
DURATION Arrival '1.475+Gamma[1.7469,8.1041]';
DURATION Service 'Normal[13.66,8.02]';

/* Queue Discipline
DISCIPLINE ClientsWait 'Service.TotInst<0.47*TotalInstances ? TimeIn : EvenOdd';

```

Figure 3.13: Stroboscope simulation script and the implementation of the discovered queue property change.

In order to check if providing the simulation model with the knowledge describing the exact timing of the queue discipline change makes a meaningful difference in simulation results, the mean waiting times of the two client types (i.e. tags with even and odd IDs) in the queue was monitored in the simulation output. Eleven independent simulation models (each with 10 replications) were run each with a 10% increment in the percentage of service instances after which the queue discipline change occurred. In other words, in the first model, it was assumed that the queue discipline was changed from FIFO to PRI after 0% of service instances were completed (i.e. PRI discipline from the very beginning); in the second model, it was assumed that the queue discipline was changed from FIFO to PRI after 10% of service instances were completed; and eventually, in the eleventh model, it was assumed that the queue discipline was changed from FIFO to PRI after 100% of service instances were completed (i.e. FIFO discipline until the end). Each simulation replication was continued until 10,000 service instances were completed and therefore, a total of 100,000 service instances were modeled for each combination of FIFO and PRI percentages. As described earlier, the designed queue data mining algorithm revealed that in the first approximately 47% of the service instances the queue discipline was

FIFO which was then switched to PRI. Therefore, one more simulation model (with 10 replications) was run in which the queue discipline was changed from FIFO to PRI after exactly 47% of service instances were completed. Figure 3.14 shows the average waiting times for even ID tags and odd ID tags in seconds, obtained from these simulation models. In this Figure, the horizontal axis shows the percentage of service instances in each simulation model after which the queue discipline is changed from FIFO to PRI, and the vertical axis indicates the average of the mean waiting times obtained from the results of 10 simulation replications. The two points corresponding to the results of the experiment with 47% FIFO and 53% PRI are distinctively marked in the plots. As shown in Figure 3.14, if the queue discipline for the entire process is PRI (i.e. 0% on the horizontal axis), the average waiting time for even ID tags (i.e. that have higher priority) is around 12 seconds, which is much less than the same value for the odd ID tags (around 81 seconds). At 47% on the horizontal axis (i.e. the time in the real experiment where queue discipline was changed from FIFO to PRI), the average waiting time for even ID tags (i.e. that have higher priority) is around 25 seconds and the same value for odd ID tags is around 63 seconds. Also, the average waiting time for all tags (both even and odd ID tags) is around 39 seconds. In comparison, if the queue discipline is FIFO from the beginning to the end (i.e. 100% on the horizontal axis), both tag types have the same average waiting times of around 38 seconds. Figure 3.14 also shows the values of the same measures observed in the real world experiment (where the queue discipline was changed from FIFO to PRI after 47% of the service instances were completed). As shown in this Figure, the observed average waiting time for even ID tags (i.e. that have higher priority) is around 10 seconds and the same value for odd ID tags is

around 81 seconds, and the observed average waiting time for all tags (both even and odd ID tags) is around 35.5 seconds.

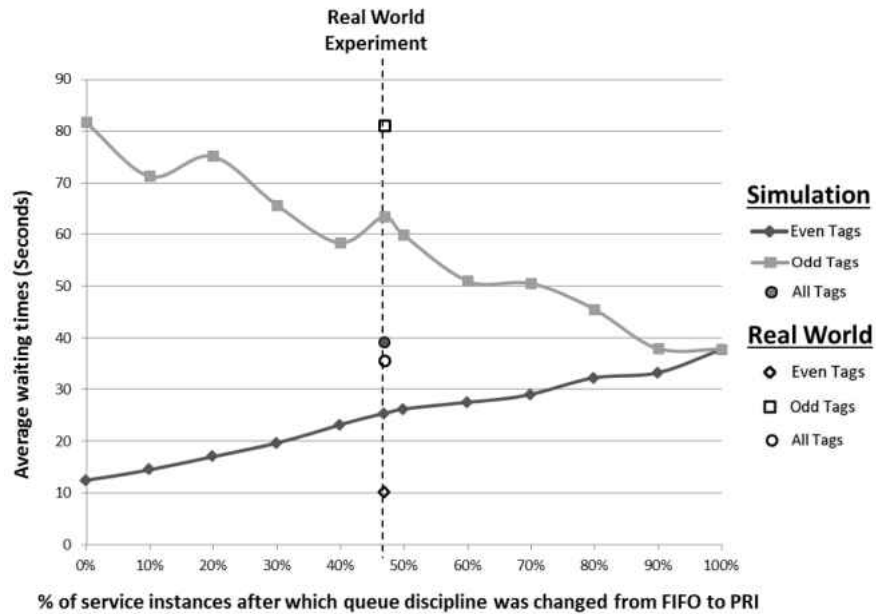


Figure 3.14: Comparison of average waiting times of even and odd ID tags.

3.7 Summary and Conclusions

Planning, control, and monitoring of construction operations in general, and heavy construction and infrastructure projects in particular, is facilitated using simulation modeling. Queuing systems are typically abstracted and represented as part of the overall project model in most simulation systems. However, in the absence of precise data with high spatial and temporal accuracy, realistic modeling of arrival and service processes in queuing systems simulation is not a trivial task. To alleviate this problem, data collection and knowledge extraction with regard to the interaction of entities in a queuing system was investigated in the research presented in this Chapter. In particular, this Chapter discussed major queue properties, and described the designed

algorithms for knowledge discovery in queuing systems. The validity and robustness of the developed methodology was assessed using a series of experiments and through collecting and mining empirical data.

The designed data collection and knowledge-discovery methodology was successful in finding the interarrival and service times, and more importantly the underlying discipline of queuing systems. A key observation was that despite the convenience of modeling interarrival and service times using the Exponential probability distribution, this distribution may not necessarily be the true representative for certain operational scenarios. In fact, the empirical analysis of results obtained from the experiments conducted in this research revealed that the Exponential distribution was in most cases the worst fit probability distribution. Although this is a legitimate finding only in the context of this study and should not be generalized to all queues, it suggests that the assumption of Exponential distribution for interarrival of entities to a queue may not be always correct. Furthermore, the designed queue discipline discovery and data mining algorithms were used in several experiments to analyze queue data and detect the predominant queue discipline, as well as any sudden changes in queue discipline during the course of the operations. In particular, three initial experiments were conducted to assess the queue properties discovery algorithms. Results indicated that these algorithms were able to successfully detect queue properties for use inside a DES model. Furthermore, in a separate experiment in which queue discipline was switched at an arbitrary point in time during the operation of the queuing system, the effect of discovering the exact point at which the queue discipline was changed on the results of the simulation was evaluated. It was found that incorporating knowledge describing a real

queuing system into a data-driven simulation model (as opposed to using static input data) can result in simulation outputs that better resemble the observations taken from the real system.

It was concluded from the validation results that relying solely on expert judgments, secondary (historical) data from past projects, and purely mathematical theories without considering the nature and unique characteristics of the current project may result in misrepresentation of the real system in a construction simulation model. This can adversely affect the reliability of the simulation output and make the results unacceptable for decision-making.

The main contribution of the research presented in this Chapter to the body of knowledge and practice is that it enables the extraction of queue properties in dynamic constantly-changing settings such as construction and infrastructure field activities. This is of practical value because queue modeling is one of the most important applications of simulation in general and DES modeling in particular. By providing factual data and extracting more reliable input knowledge such as queue properties for a simulation model, the output of the model is more representative and reliable.

CHAPTER 4: CONSTRUCTION EQUIPMENT ACTIVITY RECOGNITION FOR SIMULATION INPUT MODELING USING MOBILE SENSORS AND MACHINE LEARNING CLASSIFIERS³

4.1 Introduction

According to the United States Department of Commerce, construction and infrastructure projects comprise a trillion dollar industry with a continuous annual increase in pace [86]. Although there have been many efforts to increase the productivity of construction and infrastructure projects in recent years, the industry is still suffering from low productivity growth [87, 88, 89, 90]. There are several key factors that can influence productivity in construction and infrastructure industry, including the uncertain, dynamic, and transient nature of most construction projects. During the pre-construction phase, and due to the lack of data, it is customary to make engineering assumptions about the availability of tools, resources, information, materials, equipment, construction methods, and flow of activities [88]. Although a level of versatility is often considered for such assumptions, the dynamics involved in most projects as they enter the construction phase, makes it necessary to revise initial project plans and decisions, which may in turn result in potential delays and rework [88, 91, 92].

As infrastructure projects increasingly become larger and more complex in nature, traditional manual quantitative analysis methods mostly fail to effectively and accurately capture key project productivity performance indicators [55]. Therefore, computer simulation models capable

³ *The materials of this Chapter have been previously published in:[85] R. Akhavian, A.H. Behzadan, Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers, Adv. Eng. Inf. (2015).*

of modeling uncertainties and stochastic events have become more relevant to the decision-making process especially when real world evaluation is difficult, expensive, or time-consuming. To achieve the best results, a simulation model should accurately represent the real engineering system through the integration of data that describe the real world resources and processes [90]. It is imperative that manual data collection techniques such as direct observations and field surveys are not efficient ways to obtain large volumes of high quality data in a timely manner [93]. Thus, automated data collection using sensors, vision-based systems, and laser scanners have gained credibility in quantitative analysis of construction activities.

Process-level data collection deals with data from construction resources (i.e. equipment, labor, material). Detailed resource activity recognition using these data has a great potential in discovering knowledge about activity durations and precedence, resource flows, and site layout. Among different types of process-level knowledge, activity duration is undoubtedly one of the most influential factors as there is always an uncertainty component to duration values that can propagate in time and/or space and consequently affect the outcome of the decision-making process [70, 94]. Therefore, a systematic approach for action recognition that leads to precise activity duration extraction can boost the accuracy of decision-making tools such as simulation models. It has been widely discussed that inaccurate and unrealistic simulation models with static input data built upon expert judgments, secondary data (from past projects), and assumptions made on the basis of available resources and information during the pre-construction phase are major impediments that prohibit the widespread use of simulation models within the construction industry [55, 95].

In an effort to address this challenge, the author has investigated the applicability of data-driven simulation for construction operations analysis [5, 96]. In the author's previous work, a wireless network of sensors attached to different articulated parts of construction equipment was designed and implemented [5, 96]. However, due to technical and practical difficulties associated with mounting sensors on construction equipment body parts (e.g. attachment and detachment of different sensors for every data collection session, construction site dust and noise) and data storage issues, a more pervasive data collection scheme is used in this study. This Chapter presents the findings on a critical component of this doctoral research, a ubiquitous data sensing and analysis system that captures multi-modal process data from construction equipment using mobile sensor nodes, and employs data mining and process reasoning methods to transform raw data into meaningful knowledge that can be ultimately incorporated into data-driven simulation models. In this Chapter, first, a comprehensive literature review is conducted to help identify the gaps in knowledge and practice, and put the presented work within proper context. Next, the requirements and necessary level of detail (LoD) and resolution in activity recognition is discussed, and the designed methodology is described. Finally, the experimental results of the developed methodology are presented and further discussion about the results is provided.

4.2 Previous Work

The framework presented in this Chapter consists of a) an activity recognition architecture using built-in smartphone accelerometer, gyroscope, and positional sensors that is used to b) detect distinct activities performed by construction equipment for c) construction simulation input

modeling. Therefore, this Section provides a comprehensive literature review in each of these three domains.

4.2.1 *Action Recognition Using Accelerometer and Gyroscope Data*

A three-dimensional (3D) accelerometer is a sensor that returns values of acceleration, and a 3D gyroscope is a sensor that returns the angular velocity about x, y, and z axes [97]. The idea of recognizing activities using accelerometers have been around since the 1990s where researchers leveraged wearable devices to report instantaneous and sudden vibrations of human targets [98, 99, 100]. More recently, the use of gyroscope for the same purpose has also attracted the attention of researchers [97, 101]. In particular, the adoption of such sensors in smartphones has facilitated the emergence of more context-aware applications.

Several fields including but not limited to computer sciences, healthcare, and sports have benefited from these Micro-Electro-Mechanical Systems (MEMS) inertial sensors [97, 101, 102, 103, 104]. For example, wireless accelerometers were used for the analysis of soccer players' movement patterns [105]. Using both accelerometer and gyroscope, Li, et al. [97] presented a fall detection algorithm capable of detecting static postures and dynamic transitions. However, they stated that more environmental and physiological information is needed to distinguish between more complex actions. In a similar study, identification of physical human activities using mobile accelerometer sensors was evaluated [106]. Motoi, et al. [107] proposed a human posture and walking monitoring system that works based on the speed of the ambulatory subjects.

Despite the prevalent use of such context-aware systems in non-engineering domains, research on their applications in engineering fields has been relatively limited. For instance, in a driving

safety application, Johnson and Trivedi [108] used accelerometers to detect, recognize, and record driving styles. In an industrial setting, Lukowicz, et al. [109] developed a system for segmenting and recognizing typical user gestures in a wood workshop using body-worn microphones and accelerometers. In a prototype experiment that was conducted in a laboratory setting, they simulated the assembly of a simple wooden object to recognize specifically-designed activities. As discussed in more detail in the next Subsection, construction jobsites have unique characteristics that may prohibit the wide application of such pervasive mobile data collection techniques. Challenges include but are not limited to the unstructured arrangement of resources (i.e. equipment, labor, material) that creates technical and practical problems for installing and calibrating sensors, as well as storage of non-structured or semi-structured data. Moreover, unexpected and intermittent events such as equipment breakdowns, adverse weather, and human crew motion irregularities can also add to the difficulty of interpreting sensory data collected from construction jobsites.

4.2.2 *Construction Resource Action Recognition*

Object recognition and tracking has been a major research direction of several ongoing efforts in the field of computer vision [110, 111, 112]. Unlike computer vision where almost all such studies target human action recognition and pose analysis, researchers in construction engineering and management (CEM) domain have applied similar algorithms mostly for vision-based construction resource recognition and tracking. For example, Brilakis, et al. [16] proposed a framework for vision-based tracking of construction entities. Their methodology requires calibration of two cameras, recognition of construction resources and identification of the corresponding regions, matching the entities identified in different cameras, two-dimensional

(2D) tracking of the matched entities, and finally calculation of 3D coordinates. This and similar vision-based approaches, although provide promising results for recognition and tracking of construction equipment, still require much computation in each one of the aforementioned steps. In another study, an image processing methodology was adopted for idle time quantification of hydraulic excavators [113]. The LoD of the framework, however, was limited to detection of only idle and busy states of a hydraulic excavator. For the purpose of learning and classification of labor and equipment actions, the concept of Bag-of-Video-Feature-Words model was extended into the construction domain [114]. This technique uses unsupervised learning for classification, and only considers frequency of feature occurrence for classification. Another vision-based framework was proposed by Rezazadeh Azar and McCabe [30] for dirt-loading cycles in earthmoving operations that depends on the location of equipment which requires the algorithm to be modified for every new jobsite. In a more recent study aimed at vision-based tracking of construction equipment activities, spatio-temporal features were classified using support vector machines (SVM) [115]. Most such vision-based approaches, however, need installation of expensive cameras on the jobsite, are sensitive to ambient light conditions, visual occlusions, and moving backgrounds, and are computationally expensive due to the high volume of video data that need to be processed and interpreted [90].

Another category of object recognition and tracking methods uses sensors to collect data from target objects (e.g. equipment, labor). Compared to vision-based techniques, this approach does not require camera installation and direct line of sight, and is less prone to ambient factors. Yet, installing individual sensors is still an implementation challenge. In the CEM domain, different classes of sensors such as global positioning system (GPS) receivers, radio frequency

identification (RFID), and Ultra Wideband (UWB) have been extensively used for productivity management, safety monitoring and control, and sustainability analysis [52, 116, 117, 118, 119, 120, 121].

4.2.2.1 Application of Accelerometer Sensors in Construction and Infrastructure

Accelerometer sensors have been previously used for bridge and structural health monitoring (SHM) [122, 123, 124, 125] to detect and analyze defects, deflections, and deformations. In addition to SHM, construction labor activities have been analyzed by Cheng, et al. [126] using a physiological status monitoring system containing an accelerometer. Construction labor activity classification was also investigated to automate the work-sampling process [127]. A case study was performed in an experimental setting where a mason's activities were classified using data collected from accelerometers attached to the mason's waist. Ahn, et al. [128] examined the feasibility of measuring operational efficiency of construction equipment using accelerometer data to classify three modes of an excavator operation: engine-off, idling, and working. Overall, their methodology performed well in classifying these three classes. However, the LoD in describing activities was limited to these three classes that could be otherwise intuitively distinguished. In another study, Akhavian and Behzadan [67] used MEMS inertial sensors for updating the content of construction simulation models as well as creating a real time animation of the stationary activities of a front-end loader by mounting the sensors on equipment's articulated parts for tilt tracking.

4.2.3 *Simulation Modeling in Construction*

Computer simulation tools customized for construction operations have been in use for almost three decades since the introduction of CYCLONE by Halpin [129]. Several other simulation tools such as UM-CYCLONE [130] and Micro-CYCLONE [131] were designed based on CYCLONE. Later on, a new generation of computer simulation software came into life that provided object-oriented capabilities. Stroboscope [51] and Symphony [2] are two examples of such modeling environments that are widely used by researchers due to their extensibility and added capabilities.

4.2.3.1 *Data-Driven Simulation Models*

Several previous attempts have been made in non-CEM domains to develop real time data-driven simulation models. Some highlights include Dynamic Data-driven Application Simulation (DDDAS) tools for emergency management, contaminant tracking, enhanced chemical process design, and dynamic traffic signal control [132, 133, 134, 135]. In a recent study, a railway simulation system was developed that employed a dynamic data-driven approach using real time measures [136]. Tannock, et al. [137] used the concept of data-driven simulation to develop models for supply chain in aerospace engineering.

However, a review of the literature within the CEM domain reveals a dearth of research in simulation modeling paradigms that can incorporate and work with input data at execution phase from field activities. The author has previously designed and successfully tested a methodology that integrated multi-modal (positional, angular, and payload) data collection, data mining and reasoning process, in order to update key parameters of construction simulation models using

field data [5, 96]. Results indicated that simulation models built upon the factual data outperformed the models created using static input data and engineering assumptions. In almost all other studies in CEM that used sensory data collection targeting simulation modeling, however, the only mode of data employed to extract process knowledge has been positional. For instance, Vahdatikhaki and Hammad [138] pursued a very similar methodology to Akhavian and Behzadan [5] for near real time simulation fine-tuning. Concrete production scheduling in a DES-based optimization system was updated in another study using GPS data streaming from vehicular onboard tracking system [139]. A real time simulation framework was proposed by Hammad and Zhang [69] to improve productivity and enhance safety considering the required spatio-temporal localization resolution. Song and Eldin [70] suggested real time tracking of construction equipment to update a dynamic simulation model for look-ahead scheduling. Although all such work provided more realistic input data for simulation models compared to the traditional techniques that use static input data, they only considered equipment location information to determine activity durations, which can potentially result in limited accuracy due to the existence of cases where data other than positional information may be necessary to describe an operation.

4.3 Level of Detail in Equipment Activity Recognition

Supervised classification of construction equipment activities requires labeling different action classes to train the learning algorithm. The LoD or resolution required to successfully identify different classes from sensory data, however, may vary for each application. For instance, different mechanical degrees of freedom (DoFs) of a piece of construction equipment may have

different levels of acceleration and/or angular velocity. In light of this, the hypothesis of this research is that data collection using built-in smartphone sensors enables activity recognition of construction equipment with appropriate LoD. Since the collected data are time-stamped, this can eventually lead to precise extraction of corresponding activity durations.

One major question in developing an activity recognition framework for simulation input modeling is what constitutes an *activity*? In other words, the extent to which each operation can be broken down (i.e. LoD) for modeling purposes defines the granularity of the activities modeled in the simulation. The significance of LoD in the context of modeling can be best seen in illustrative examples that use simulation results to create realistic replicas of engineering operations in visual environments such as simulation-based virtual or augmented reality [140, 141, 142]. In such environments, activities should be broken down to the most detailed level possible in order to render a smooth animation of the simulated operation. Consequently, if the final LoD does not include all mechanical DoFs, the resulting visualized scene appears unrealistic.

The state of a given piece of construction equipment can be broken down into further detailed actions. Here, action is defined as any process-level state of equipment that produces a distinctive sensory signal pattern. For example, Figure 4.1 depicts a hierarchy of actions that can be performed by a front-end loader, a widely used construction equipment. As shown in this Figure, activities of a front-end loader can be broken down into different actions based on the defined LoD. In the coarsest breakdown (i.e. level 1), 2 classes are defined: Engine Off, and Engine On. Using the definition of action above, since these two classes produce two different

sensory signal patterns, they can be treated as separate actions. In the next level, the Engine On class is further divided into the Idle and Busy classes. Therefore, the total number of possible classes in this LoD is 3 (i.e. Engine Off, Idle, and Busy). The action breakdown is continued to level 4, in which 5 classes are defined. If needed, this process can be continued even further. For example, Moving class may be divided into two subclasses of Going Forward and Going Backward. In most cases, the end application (purpose) of the activity recognition process can help determine the number of levels to which action breakdown should be continued.

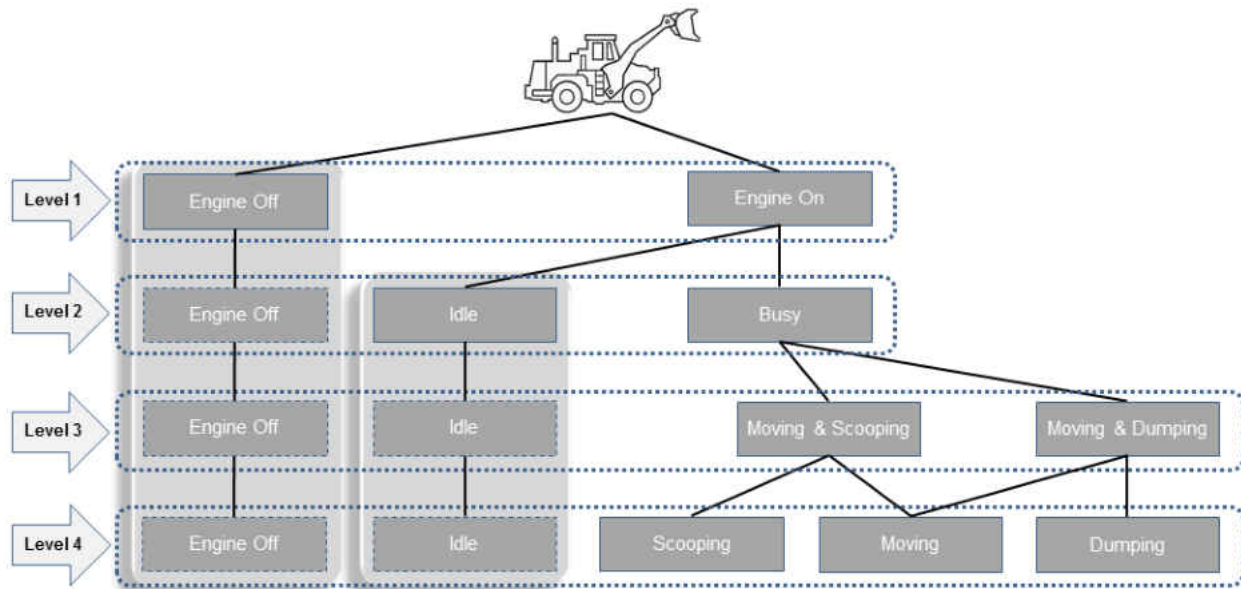


Figure 4.1: LoD in activity breakdown of a front-end loader.

As previously stated, the main focus of the research presented in this Chapter is to precisely extract activity durations. The occurrence of any action shown in an arbitrary level (referred to herein as level l) of the hierarchy of Figure 4.1 can imply that the parent action right above it (in level $l - 1$) in the tree is occurring. However, depending on the circumstances, there may be two different interpretations. For instance, if the required LoD is level 4, given that an instance of

Dumping action with duration t_1 and an instance of Moving action with duration t_2 are occurring, it can be concluded that two separate instances of Moving and Dumping action, with durations of t_1 and t_2 are taking place. On the other hand, if the required LoD is level 3, knowing that Moving and Scooping and Moving and Dumping actions are taking place with durations t_3 and t_4 , respectively, one should add up t_3 and t_4 to calculate the duration of a single instance of Busy state (e.g. dumping soil into a hauler) as t_3+t_4 . In any case, as a general rule, it is possible to derive the duration of actions in level $l - 1$ given the duration of actions in level l , and not necessarily vice versa.

Although it is desirable to have as many levels as possible when describing equipment activities, depending on the structure of activity breakdowns, the performance of activity classifications and further, activity duration extraction methodologies varies over different levels due to three important reasons. Consider Figure 4.1 as an example:

1. Training a classifier with a fixed dataset to distinguish between combined and coarser classes with similar movement characteristics (e.g. level 2) is expected to be more successful than dividing the same dataset into single more detailed classes (e.g. level 4). This is mainly because the more are the levels, the less will be the number of training data points in each class. Moreover, dividing some of the actions to more detailed actions and creating new classes may result in having imbalanced data. For example, if in a given operation, the breakdown of the collected data points in level 3 is 30% Engine Off, 30% Idle, 20% Moving and Scooping, and 20% Moving and Dumping, then in level 4, the number of data points in the last two activities will be further broken down into smaller

portions, whereas other states such as Idle still have more data points for training the classifier.

2. Signal patterns to which classification is applied are expected to become more similar in each class when going down in the tree; meaning that for example, Scooping, Dumping, and Moving actions are more likely to have similar signal patterns within their individual classes than Moving and Scooping or Moving and Dumping. This makes it more difficult for the classifier to distinguish between different classes.
3. Automated extraction of activity durations from classified activities is performed by algorithms to detect separate instances of equipment activities and calculate the durations based on the data segmentation (i.e. window sizing). For example, if 10 segments in a row were labeled by the predictor model as the activity Scooping, but 1 or 2 segments are labeled Dumping, and right after them, again 8 segments are labeled as Scooping, the algorithm ignores the 1-2 Dumping segments and count them as Scooping, resulting in around 20 segments labeled as Scooping. Therefore, the accuracy of the activity classification algorithm (or trained model) which is defined by the ratio of correct predicted labels over actual labels in each class will be slightly different from the accuracy of activity duration extraction algorithm. This difference is obviously higher when having many classes with shorter durations rather than fewer classes (combined classes) with longer durations.

As a result, although a higher LoD is more desirable, the classifier may not necessarily perform well in the lower levels given the relatively large number of classes. This trade-off indicates that there should be an optimal level in the action hierarchy where a balance exists between the

number of classes and the accuracy of the classification process and duration extraction. This issue will be further examined in this Chapter. It must be noted that Figure 4.1 serves only as a demonstration example and a motivation case and each particular operation may require that a different action LoD hierarchy be constructed to best determine the relationship between the duration of child and parent actions for a specific operation. Likewise, the selection of the optimal point, where the number of classes vs. the LoD results in the best performance, may vary from case to case.

4.4 Activity Recognition Methodology

The general architecture of the designed framework is depicted in Figure 4.2. In this methodology, multi-modal data is collected from different sensors (i.e. accelerometer, gyroscope, GPS) embedded in mobile (smartphone) devices placed inside construction equipment cabins. While GPS data is used later on to provide additional contextual information such as the proximity of two pieces of equipment (e.g. a front-end loader and a hauler) or work zone vicinity approximation [5], for accurate duration extraction (the focus of this study), mainly accelerometer and gyroscope data are subject to a major data processing effort. In particular, after collecting raw data, specific features should be extracted for classification. However, not all such features may contribute to the classification process, and thus a feature selection step needs to be taken. Selected features go through the training process and then new actions are recognized at the LoD specified in the training phase. Each one of these steps is described in detail in the following case study where real world data was used.

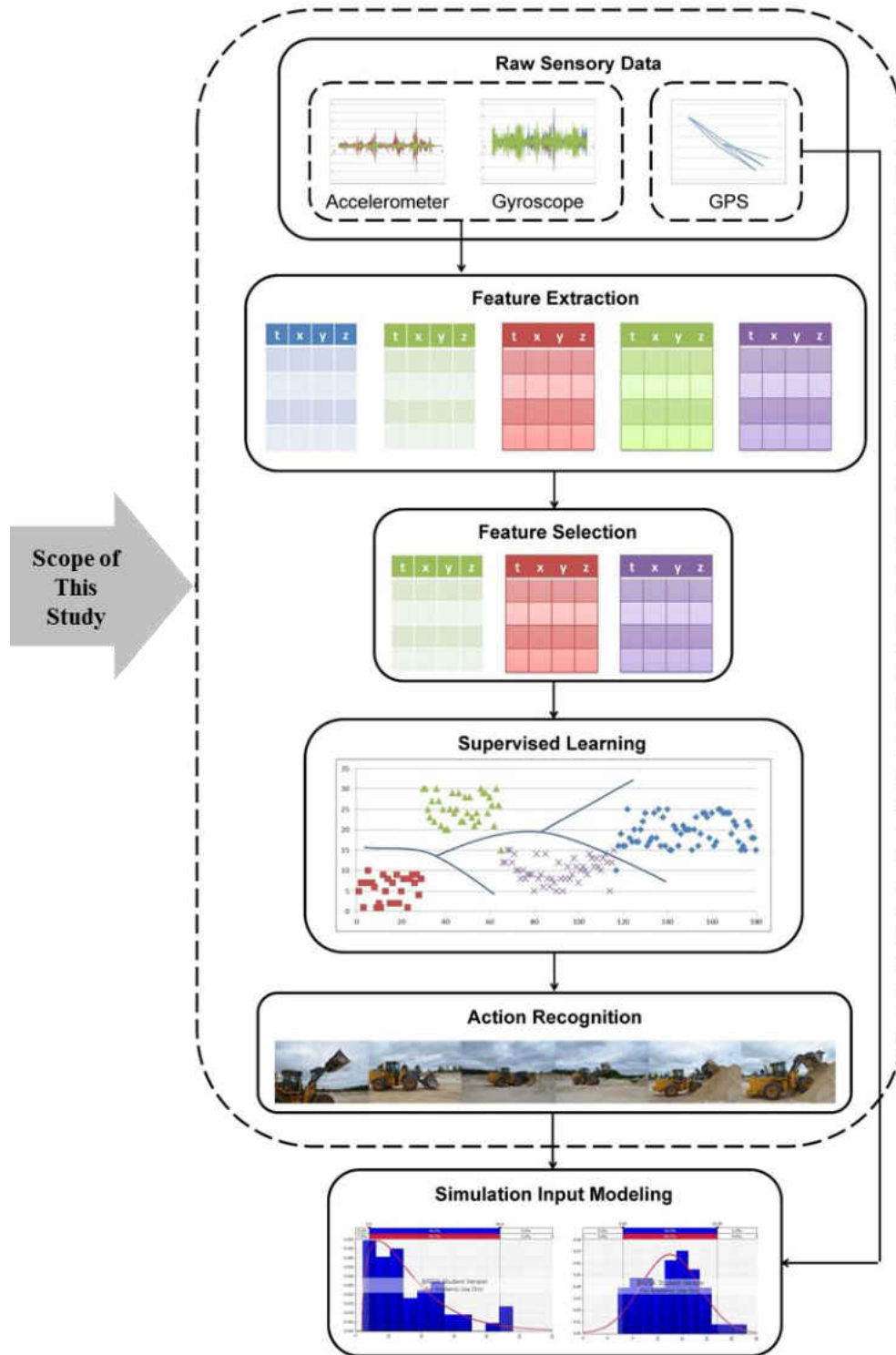


Figure 4.2: Developed system architecture for simulation input modeling.

4.4.1 *Experiment Setup*

In all experiments conducted in this study, smartphones were placed inside the equipment cabin for data collection. For each scenario, two smartphones were simultaneously used to guarantee the uninterrupted storage of data. It must be noted that since data collection and feature extraction is done using tri-axial data, results do not depend on the placement orientation of the data collection device. Moreover, potential significant correlation between each pair of axes is reflected in three of the extracted features, thus guaranteeing capturing any distinguishable feature related to the placement orientation of the data collection devices. In order to fully automate the process of data collection, low-cost near field communication (NFC) RFID smart tags were also used [143]. NFC tags were glued to the device holder (i.e. suction cup attached to the side window of the cabin) to automatically launch the data logger application once the smartphone was placed in the holder. A JOHN DEERE 744J front-end loader was employed for data collection. All experiment operations were fully videotaped for later activity annotation and labeling, and visual validation. Figure 4.3 shows how data collection devices were mounted and secured inside the target equipment cabin.



Figure 4.3: Smartphones mounted inside the front-end loader cabin.

4.4.2 *Data Collection and Logging*

Data was collected using commercially available data logger applications for iOS and Android devices. The sampling frequency was set at 100 Hz. Among different modes of data collected in this study, it was observed that acceleration (i.e. vibration) values resulted from different equipment motions had the highest degree of volatility. Several sensor manufacturers have recommended that a bandwidth of 50 Hz be used for normal-speed vibration and tilt sensing applications. Therefore, in this research, and considering the Nyquist criterion in signal processing [144], the sampling frequency was set at twice this value or 100 Hz. This bandwidth guaranteed that no significant motion was overlooked and at the same time, the volume of recorded data was not prohibitively large. Data was stored with comma separated value (CSV) format for processing in Microsoft Excel. The logger applications provided time-stamped data which facilitated the synchronization of data and video recordings. As mentioned earlier, GPS data is not directly used in data mining processes employed in this study and was only collected to demonstrate the potential of acquiring high accuracy positional data for such context-aware applications. Figures 4.4 visualizes part of the collected accelerometer, gyroscope, and GPS data.

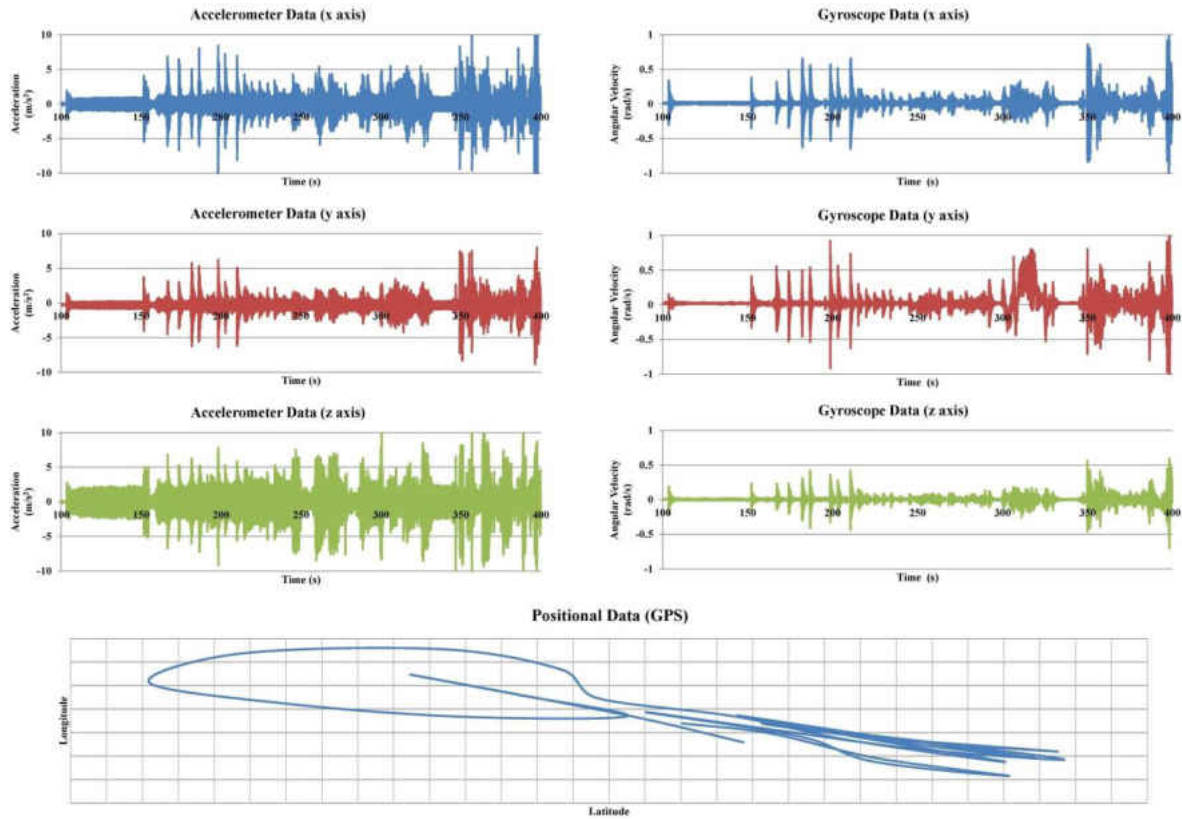


Figure 4.4: Snapshots of collected sensory data.

4.4.3 Data Processing

4.4.3.1 Feature Extraction

Raw data must be first represented in terms of specific features over a window of certain data points. In this research, mean, variance, peak, interquartile range (IQR), correlation, and root mean error (RMS) are the statistical time-domain features that were extracted from data. Moreover, signal energy was picked as the only frequency-domain feature since it had already shown positive discrimination results in previous studies [145, 146] for context recognition using accelerometer data. These 7 features were extracted from both accelerometer and gyroscope data corresponding to each of the x, y, and z axis. Since both sensors return tri-axial values (x, y, z), a

total of 42 (i.e. multiplication of 7 features from 2 sensors in 3 axes) features were extracted. The size of the window depends on the sampling frequency and thus, varies for different applications. However, it should be selected in such a way that no important action is missed. This can be achieved by overlapping consecutive windows. Previous studies using accelerometer for context recognition have suggested a 50% overlap between windows [72, 128, 147]. Time-domain features can be extracted using statistical analysis. However, the frequency-domain feature (i.e. signal energy) should be extracted from the frequency spectrum which requires signal transformation. In this study, fast Fourier transform (FFT) was used to convert the time-domain signal to the frequency-domain. In order to be computationally efficient, FFT requires the number of data points in a window to be a power of 2. Data was initially segmented into windows of 128 data points with 50% overlap. Therefore, given a sampling frequency of 100 Hz, each window contained 1.28 seconds of the experiment data. A sensitivity analysis presented in Section 4.5 provides more detail about the process of selecting the proper window size. The entire data analysis process including feature extraction was performed in Matlab.

4.4.3.2 Feature Selection

Feature selection is the process of picking a subset of originally extracted features to optimally reduce the feature space [148]. In other words, among all extracted features, there are some that may not add to the accuracy of the classification. This might be due to the correlation that exists among the collected data and consequently extracted features, since many actions result in a similar pattern in different directions and/or different sensor types (i.e. accelerometer vs. gyroscope). Therefore, in order to reduce the computational cost and time of the classification process, and increase its accuracy, a subset of the discriminative features is selected by filtering

out (removing) irrelevant or redundant features [149]. In this study, two filtering approaches are used: ReliefF and Correlation-based Feature Selection (CFS). ReliefF is a weighting algorithm that assigns a weight to each feature and ranks them according to how well their values distinguish between the instances of the same and different classes that are near each other [148]. CFS is a subset search algorithm that applies a correlation measure to assess the goodness of feature subsets based on the selected features that are highly correlated to the class, yet uncorrelated to each other [150].

Using CFS, irrelevant and redundant features were removed which yielded 12 features (out of 42). These features were then ranked by ReliefF using their weight factors. The first 12 features selected by ReliefF were compared to those selected by CFS and the 7 common features in both methods were ultimately chosen as the final feature space. Table 4.1 shows the selected features by each filter as well as their intersection.

Table 4.1: Selected features by CFS and ReliefF and their intersection (A: Accelerometer, G: Gyroscope).

Filter	Selected Features	Common Selected Features
<i>CFS</i>	A_mean_x, A_mean_y, A_mean_z, A_peak_x, A_iqr_y, A_iqr_z, A_correlation_z, A_rms_z, G_mean_x, G_mean_y, G_mean_z, G_variance_x	G_mean_z A_mean_x G_mean_x A_mean_y A_mean_z A_iqr_z
<i>ReliefF</i>	G_mean_z, A_mean_x, G_mean_x, A_peak_z, A_mean_y, A_correlation_y, A_correlation_x, A_mean_z, A_iqr_z, A_peak_x, A_peak_y, G_rms_z	A_peak_x

4.4.3.3 Supervised Learning and Classification

A learning algorithm can be supervised or unsupervised depending on whether or not different classes are labeled for training. Although unsupervised methods can be employed for equipment action recognition [114], supervised learning algorithms provide better performance for this purpose [115]. This is mainly due to the fact that action classes of a piece of equipment consist of some classes with limited number of instances. This creates an imbalanced set of classes (caused by large differences between the number of instances in some classes) that can very likely lead to over-fitting in unsupervised learning classification. Among several supervised learning methods those that follow more complex algorithms may seem more accurate in classification. However, the choice of the learning algorithm is highly dependent on the characteristics and volume of data. As a result, a “single” best classifier does not generally exist and each case requires unique evaluation of the learning algorithm through cross validation [151]. Therefore, a number of learning algorithms are tested in this research to compare their performance in classifying actions using sensory data.

As per the discussion of LoD in breaking down the activities in Section 4.3, in this experiment, classification was performed by labeling the classes in different LoDs. Following the same hierarchy of actions presented in Figure 4.1, and starting from level 2 (since level 1 is too coarse for the purpose of this study) the first set of training and classification algorithms is applied to three classes namely Engine Off, Idle, and Busy. Next, the Busy class is broken down into two subclasses of Moving and Scooping, and Moving and Dumping, and so on for level 4.

For action classification, five supervised learning methods were used: 1) Logistic Regression, 2) K-Nearest Neighbor (K-NN), 3) Decision Tree, 4) Neural Network (feed-forward backpropagation), and 5) SVM. Using different classifiers reduces the uncertainty of the results that might be related to the classification algorithm that each classifier uses.

4.5 Results and Discussion

Starting from level 2, for each LoD, five classifiers were trained. Training and testing were performed through stratified 10-fold cross validations. In a k -fold cross validation the dataset is divided into k sets of equal sizes, and classifiers are trained k times, each time they are tested on one of the k folds and trained using the remaining $k - 1$ folds. Moreover, in the stratified k -fold cross validation, each fold contains almost the same proportions of classes as in the whole dataset. The mean accuracy is reported as the accuracy of each class. Result of the classification performance for each case (i.e. LoD) is presented in Table 4.2 in terms of overall classifier accuracy. As shown in Table 4.2, Neural Networks had the best relative overall accuracy among all five classifiers in all the LoDs. Moreover, although in level 2 with 3 classes the accuracy gets to as high as 98.59%, the highest accuracy in level 3 with 4 classes is 81.30% which is less than that of level 4 with 5 classes, which is 86.09%.

Table 4.2: Overall accuracy of classifiers for each LoD.

	Classifier	Accuracy (%)
Level 2	ANN	98.59
	DT	97.40
	KNN	97.65
	LR	96.93
	SVM	96.71
Level 3	ANN	81.30
	DT	81.21
	KNN	80.51
	LR	77.58
	SVM	78.03
Level 4	ANN	86.09
	DT	73.78
	KNN	84.20
	LR	84.42
	SVM	78.58

As stated earlier, construction equipment activity recognition has been previously explored through vision-based technologies. Gong, et al. [114] reported an overall accuracy of 86.33% for classification of three action classes of a backhoe. In a more recent study, Golparvar-Fard, et al. [115] achieved 86.33% and 76.0% average accuracy for three and four action classes of an excavator, respectively, and 98.33% average accuracy for three action classes of a dump truck. Although the target construction equipment are different in each case and action categories varies in these studies, the developed framework in this study that uses IMUs for the first time for construction equipment action recognition shows promising results when compared to existing vision-based systems that have been the subject of many research studies in the past few years.

Prior to conducting a detailed analysis of results, one more sensitivity analysis was performed to confirm that the intuitively selected window size of 1.28 seconds is actually the best option. Considering that FTT uses windows that are sized as a power of 2, window sizes of 0.64 seconds and 2.56 seconds were selected for this sensitivity analysis. A window size smaller than 0.64 seconds (e.g. 0.32 seconds) is too small, while a window size larger than 2.56 seconds is too large given the type of activities observed in the experiment from the annotated videotaped data. The sensitivity analysis was performed for the best classifier (i.e. the Neural Networks) and the highest number of classes (i.e. 5 classes) that required the most computation and has the least numbers in some classes. Table 4.3 shows the result of the sensitivity analysis. According to Table 4.3, a window size of 1.28 seconds that corresponds to 128 data points has the best accuracy among the all three window sizes and thus is used for further analysis.

Table 4.3: Accuracy of the trained neural networks for different window sizes.

Window Size (Sec.)	0.64	1.28	2.56
Accuracy (%)	82.03	86.79	82.45

Next, confusion matrices for classification performance for each class within a LoD are constructed. In a confusion matrix, row labels represent actual (real world) classes while column labels represent predicted classes (from sensory data patterns). Therefore, the percentage value shown in each cell in a row indicates the accuracy of the classifier in identifying the class corresponding to that row. With the same token, diagonal elements of the matrix represent classes that were classified correctly (predicted vs. actual), while non-diagonal elements

represent misclassified instances. Table 4.4 shows the number of segments of each activity within each class and the number of instances in each equipment action category.

Table 4.4: Number of segments and instances for each activity within each class.

Classification	Activity	Number of Segments	Number of Instances
<i>Level 4 (5 classes)</i>	Engine Off	55	2
	Idle	36	2
	Scooping	32	4
	Moving	180	8
	Dumping	57	4
<i>Level 3 (4 classes)</i>	Engine Off	55	2
	Idle	36	2
	Moving & Scooping	142	4
	Moving & Dumping	127	4
<i>Level 2 (3 classes)</i>	Engine Off	55	2
	Idle	36	2
	Busy	269	1

Figures 4.5 through 4.7 show the confusion matrices for classification of classes in levels 2, 3, and 4, with 3, 4, and 5 classes, respectively, using Neural Networks.

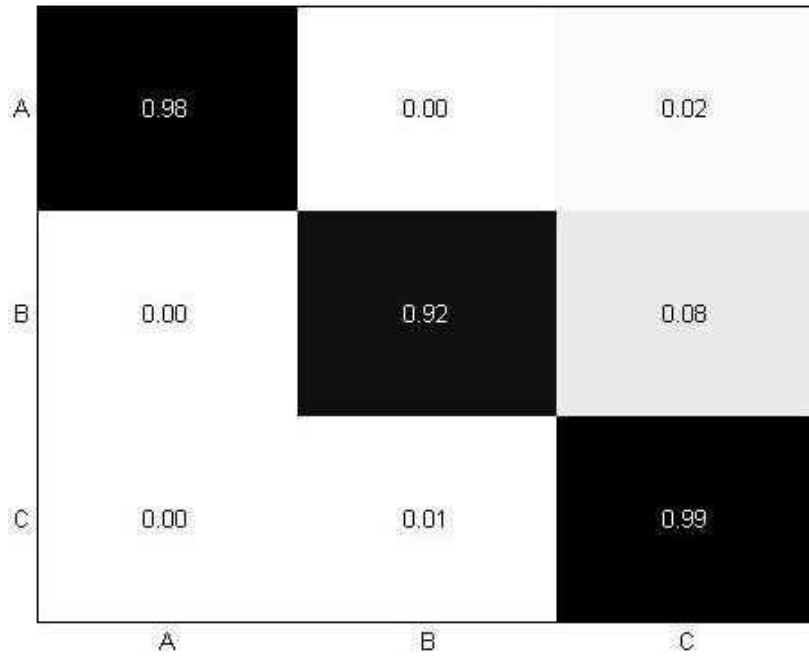


Figure 4.5: Neural network confusion matrix for level 2 (A: Engine off, B: Idle, C: Busy).

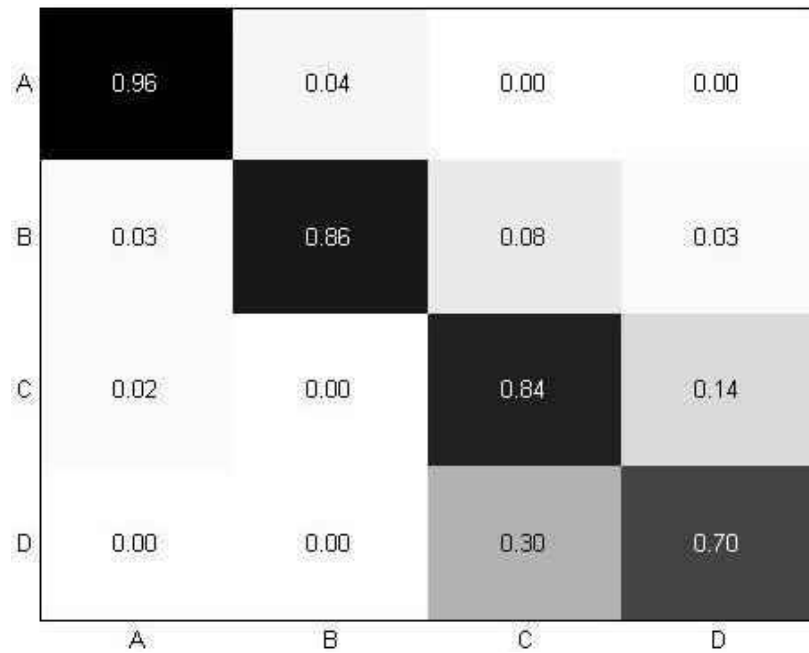


Figure 4.6: Neural network confusion matrix for level 3 (A: Engine off, B: Idle, C: Moving and Scooping, D: Moving and Dumping).

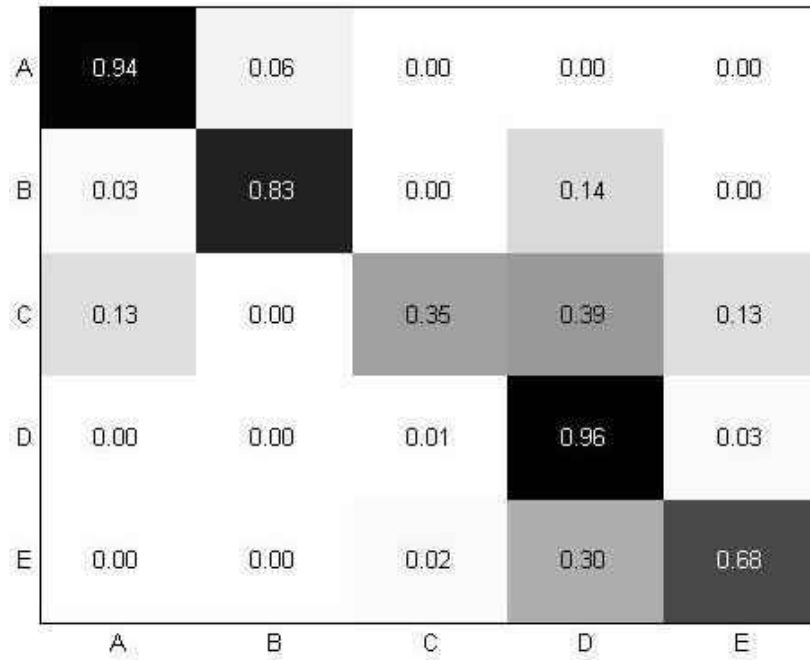


Figure 4.7: Neural network confusion matrix for level 4 (A: Engine off, B: Idle, C: Scooping, D: Moving, E: Dumping).

Following activity recognition and classification, activity durations should be extracted for simulation input modeling. Detected instances of each activity have a certain number of classified windows. Since each window is 1.28 seconds with 50% overlap with the previous window (i.e. 0.64 seconds), the duration of each instance is calculated using the Equation 4.1,

$$t(\text{seconds}) = (n+1) \times 0.64 \quad (4.1)$$

in which n is the number of detected windows of each class in each instance, and t is the duration of that instance of the target class. In order to find the LoD which results in the most accurate activity duration extraction, normal root mean squared errors (NRMSEs) of the actual activity durations and extracted ones are calculated and tabulated in Table 4.5.

Table 4.5: NRMSEs of extracted activity durations compared to the real activity durations.

Classification	Activity	NRMSE of Durations (%)
<i>Level 4 (5 classes)</i>	Engine Off	2.9
	Idle	4.5
	Scooping	44.1
	Moving	6.5
	Dumping	23.4
<i>Level 3 (4 classes)</i>	Engine Off	2.6
	Idle	4.4
	Moving & Scooping	3.8
	Moving & Dumping	5.5
<i>Level 2 (3 classes)</i>	Engine Off	2.5
	Idle	3.6
	Busy	0.0

Since the Busy state recognized as one of the classes in level 2 is not an actual activity, it is not considered in the discussion of activity durations extraction. However, for levels 3 and 4, as shown in Table 4.5, the NRMSE results indicate that in level 3 with 4 classes, the extracted activity durations were more accurate than those in level 4 with 5 classes. This is while classification accuracy was higher in level 4. There are two important factors that explain these results. First, combining multiple classes to build a single coarser class of activity for simulation input modeling (i.e. extracting activity durations) should be in a way that different classes that

are combined to form a new class are adjacent to each other, thus summing the activity durations is meaningful when combining them. For example, from level 4 to level 3, the classes Scooping and Dumping cannot be combined because they did not actually occur following each other (i.e. a loader first scoops, then moves, then dumps) and thus, their durations cannot be added up. Therefore, the classification accuracy may not be as high as the lower detailed level since now two different activities (with some different signal patterns) are combined. Second, classification accuracy per se is not sufficient to know what is the best combination of classes for activity duration extraction, because there might be a few misclassified windows within each detected instance that although affect the classification accuracy negatively, but are ignored in duration extraction because they last as short as one or two window sizes (0.64 or 1.28 seconds) in an instance that is on average 10 seconds. For example, the observed average of dumping activity in the conducted experiments was around 10 seconds and in an instance, two consequent windows were labeled as scooping. Therefore, the algorithm for extracting durations easily ignores 1.92 seconds of scooping in the middle of 10 seconds of dumping. As a result of these two important issues, the classification accuracy may not always be the same as the activity duration extraction accuracy. This was observed in the experiments conducted with different number of classes. Therefore, such a thorough study is important to understand the theoretical and working principles of the framework targeting simulation input modeling using activity recognition and classification.

4.6 Summary and Conclusions

The goal of the research presented in this Chapter was to investigate the prospect of using built-in smartphone sensors as ubiquitous multi-modal data collection and transmission nodes in order to detect detailed construction equipment activities. The discovered process-level knowledge can provide a solid basis for different applications such as productivity improvement, safety management, and fuel use and emission monitoring and control. In addition, this methodology can serve as a basis for activity duration extraction for the purpose of construction simulation input modeling. A case study of front-end loader activity recognition was used to describe the methodology for action recognition and evaluate the performance of the developed system. In doing so, several important technical details such as selection of discriminating features to extract, sensitivity analysis of data segmentation window size, and choice of classifier to be trained were investigated.

In summary, results indicated that different equipment actions generate distinct data patterns (i.e. signatures) in accelerometer and gyroscope data. In particular, using smartphone built-in sensors demonstrated a perfect success (i.e. classification accuracy of over 98%) in recognizing the engine off, idle, and busy states of construction equipment. This can be the basis of future studies targeting automated state recognition of construction equipment for sustainability and safety purposes. Careful examination of the classification confusion matrices in the highest LoD showed that the classification of activities was successfully performed (i.e. around 90% classification accuracy) in detecting some classes such as engine off, idle, and moving, whereas in activities such as dumping, and scooping, lower accuracies were achieved. It is worth

mentioning that the classes that have higher classification accuracy have more distinctive vibration and angular velocity features as well. Therefore with more sensing devices and technologies, the results can improve even further. Having said that, as indicated in Table 4.5, the lower classification accuracy for some levels/classes does not necessarily translate into a low accuracy in activity duration estimation, which is the ultimate goal of integrating process sensory data for activity classification and creating more accurate simulation input models.

Another key contributing factor to classification performance is what was referred to in Section 4.3 as the problem of having imbalanced data for classification. It was observed that performance is much better when dealing with better balanced data. The results of this study compared the performance of different classifiers, number of classes, and window sizes in recognizing construction equipment activities. However, as discussed in the Section 4.5, proper attention should be paid to the fact that the accuracy of activity duration extraction is contingent upon a variety of interconnected factors.

Some of the directions for future work on this topic include the investigation of algorithms that suit classification of imbalanced data. To this end, a number of methodologies that can potentially handle this situation, including under-sampling and over-sampling [152], and cost-sensitive analysis (i.e. giving weights to data according to the number of data points) [153] can be explored. In addition, it was observed in the case study that some classifiers showed a better performance in classifying certain classes. Therefore, one possible solution to improve the overall classification accuracy is to use multiple classifiers in conjunction with one another. To

this end, ensemble methods [154] and meta-learners [155] can be used to combine different classifiers.

CHAPTER 5: SMARTPHONE-BASED ACTIVITY RECOGNITION AND CLASSIFICATION OF CONSTRUCTION WORKERS

5.1 Introduction

Effective and timely analysis and tracking of workforce activities is essential to overall productivity measurement, progress evaluation, labor training programs, and safety and health management in construction projects. Activity analysis is performed to evaluate the time spent on interconnected construction tasks involving labor force. Such analysis includes monitoring, benchmarking, and improving the amount of time spent on unique processes involved in typical construction activities [156]. The outcome of this analysis can be used for stochastic simulation input modeling, work sampling, and integrated detailed assessment and continuous workflow improvement.

Process monitoring and control provides a solid basis for tracking and measurements required for activity analysis. Recent advancements in automated data collection to track resources and measure work progress have shown promising prospects for streamlining crew activity analysis compared to the conventional (manual) approaches such as direct observations and survey-based methods. This is mostly because manual methods involving human observers are tedious, time consuming, and error-prone. Furthermore, large amounts of data should be collected in order to maintain the statistical significance of observations.

However, automated technologies for data acquisition are still being assessed in terms of their reliability and feasibility in construction domain applications. In one hand, vision-based

techniques have been proposed and investigated by a number of researchers for automated activity analysis [157]. On the other hand, wireless sensor-based methodologies have been examined to collect spatio-temporal activity data [156]. The first set of investigations is prone to extant occlusions and illumination variability in construction jobsites. Alternatively, the second group does not require clear line of sight and extensive computations and can provide low cost solutions compared to laser-scanner for instance. A longstanding challenge and impediment to the widespread use of sensor-based data collection schemes is that traditional sensor installation and maintenance in construction jobsites is not a trivial task (if not at all impossible) due to ambient factors such as dust, adverse weather conditions, and harsh working environments.

To remedy this situation, a relatively newer data collection technique has been trending which uses ubiquitous sensors that are readily available to and carried by most individuals on a daily basis. Such technologies are, for instance, provided through built-in sensors in most mobile phones. Mobile devices are advantageous over other activity recognition data collection platforms since they unobtrusively provide a self-sufficient data collection, computing, and storage scheme. Recently, many construction research studies have taken advantage of ubiquity of smartphones to design and prototype useful applications for construction workers on the jobsite [158, 159]. Such applications in essence deliver information to the site personnel, while there is a great potential to infer information using the built-in sensors. A typical smartphone has an almost inclusive subset of these context-aware sensors including accelerometer, gyroscope, GPS, magnetometer, barometer, proximity sensors, light sensors, Bluetooth, Near Field Communication (NFC), and cameras [158]. Recently, construction equipment and tool

manufacturers have started to produce rugged drop-proof, and dust- and water-resistant smartphones specifically designed for construction jobsites (ENR 2013).

This Chapter presents a thorough evaluation of the performance of an activity analysis framework for recognition and classification of various construction worker activities using smartphone built-in sensors. In this research, data are collected from a variety of construction activities performed by construction workers and are annotated for feature extraction to train machine learning classifiers. Data-driven methodologies in activity recognition fall into one of the two major categories of generative or discriminative approaches. While in generative approach probabilistic models such as Bayesian network are used to build a description of input, the discriminative approach models the mapping from inputs to outputs or data to activities [160]. Using generative models such as hidden Markov models (HMM) and dynamic Bayesian network (DBN) is not within the scope of this research since they are not capable of capturing transitive dependences of the observations due to their very strict independence assumptions. Moreover, HMM needs significant training data to be able to detect all possible observations sequences [160]

5.2 Literature Review

5.2.1 *Automated recognition of construction worker activities*

Previous research for activity recognition and classification of construction workers mainly falls into the vision-based category. Microsoft Kinect, for example, was employed by some researchers for vision-based activity recognition in indoor and controlled environments [161,

162]. In another set of studies, 2D videos are used to collect visual data for action recognition in construction sites. For example Peddi, et al. [163] used a wireless video camera to extract human poses from video to recognize construction workers' actions. In a different study, 3D range image camera was used for tracking and surveillance of construction workers for safety and health monitoring [163, 164]. Gonsalves and Teizer [164] indicated that if their proposed system is used in conjunction with ANN, the results would be more robust for prevention of fatal accidents and related health issues. In their study on construction workers' unsafe actions, Han and Lee [165] developed a framework for 3D human skeleton extraction from video to detect unsafe predefined motion templates. All of these frameworks, although presented successful results in their target domain, require installation of multiple cameras (up to 8 in some cases), have short recognition distance (maximum of 4 meters for Kinect) and require direct line of sight for implementation. Such shortcomings have served as a major motivation to investigate alternative solutions that can potentially alleviate these problems.

Recently, researchers in construction engineering and management (CEM) have investigated the applications of sensor-based worker activity analysis. For example, a data fusion approach using ultra-wide band (UWB) and Physiological Status Monitors (PSMs) for productivity [156] and ergonomics [166] analysis was proposed. In this study, UWB and PSM data were fused and the result was categorized using a spatio-temporal reasoning approach. However, the level of detail in recognizing the activities was limited to identification of traveling, working, and idling states of workers and could not provide further insight into identified activities. Prior to this study, the integration of UWB, payload, and orientation (angle) data with spatio-temporal taxonomy-based reasoning was adopted by the author for construction equipment activity analysis to support

process visualization, remote monitoring and planning, and knowledge-based simulation input modeling [5, 67, 142]. Joshua and Varghese [127] were among the first researchers who explored the application of accelerometer in construction for work sampling. However, the scope of their work was limited to only a single bricklayer in a controlled environment. Moreover, their proposed framework used accelerometer as the sole source of motion data. Also, the necessity of installing wired sensors on the worker's body may introduce a constraint on the worker's freedom of movement.

5.2.2 Activity recognition using cellphone sensors

Detection and classification of human activities using wearable inertial measurement units (IMUs) consisting of accelerometer and gyroscope gained traction among computer science researchers in mid-2000's with applications in different fields such as healthcare and sports [103, 104, 167]. In all such studies, data pertaining to human physical movements are captured using IMUs and different postures and dynamic transitions are detected by training classifiers. However, more recent studies are geared toward leveraging the ubiquity, ease of use, and self-sufficiency of mobile phones for human activity recognition [106, 168, 169, 170]. In one study, Reddy, et al. [171] used decision tree and dynamic hidden Markov model (DHMM) to classify activities such as standing, walking upstairs, biking, driving a car, and jumping using accelerometer and GPS data. In another research, Sun, et al. [169] used support vector machines (SVMs) to build a human daily physical activity recognition system using mobile phone accelerometers. More recently, mobile phone gyroscope has been also employed in addition to accelerometer for activity recognition. For example, using accelerometer and gyroscope data and hierarchical SVM, Kim, et al. [172] classified daily activities to sitting, walking up- and

downstairs, biking, and having no motion. Moreover, Martín, et al. [173] used decision table, decision tree, and naïve Bayes to classify data from various smartphone sensors such as accelerometer and gyroscope to classify daily activities into standing, sitting, jogging, and walking upstairs.

Despite its great potential for construction automation, and considering the existing interest in and attention to recognition construction workers, the application of such emerging data collection platforms has not been fully investigated within the CEM domain. In the research presented in this Chapter, signature patterns observed in the signals received from wearable IMUs of ubiquitous smartphones are analyzed to recognize activities performed by different construction workers.

5.3 Research Objectives and Contributions to the Body of Knowledge

As stated in Section 5.2, previous research on activity recognition within the CEM domain has primarily focused on vision-based systems while a very limited number of studies aimed at developing multi-modal sensor-based data collection schemes. Hence, the presented study in this Chapter contributes to the body of knowledge by investigating construction worker activity recognition through (1) using the sensors embedded in mobile phones to (2) identify complex activities that consist of more than one task by (3) deploying combined features of accelerometer and gyroscope (i.e. IMU) data. In particular, this research provides new insight into the accuracy of recognizing construction workers' complex and continues activities through different learning algorithms (objective 2) where more than one task is performed by a worker, using mobile built-in IMUs (objectives 1 and 3).

5.4 Methodology

In this study, data are collected using mobile phone accelerometer and gyroscope sensors. Collected raw sensory data are segmented into windows containing certain number of data points. Next, key statistical features are calculated within each window. Furthermore, each segment is labeled based on the corresponding activity class performed at the time identified by the timestamp of the collected data. In order to train a predictive model, five classifiers of different types are used to recognize activities performed in the data collection experiments. Figure 5.1 depicts the steps from data collection to activity recognition.

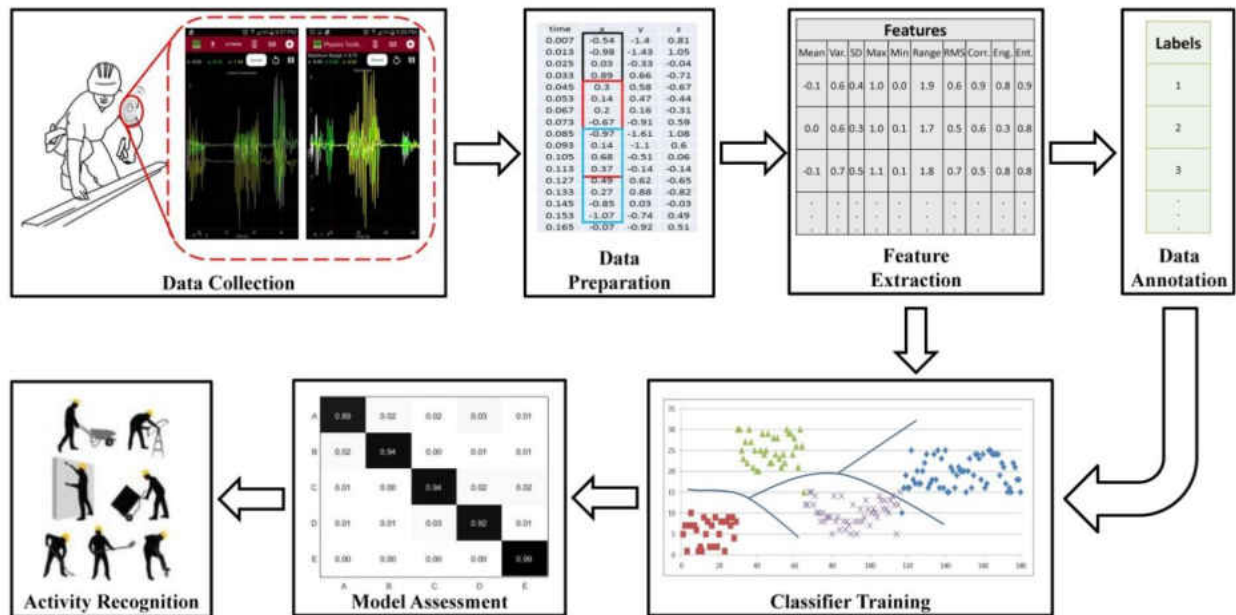


Figure 5.1: Framework for construction worker activity recognition using mobile sensors.

5.4.1 Data Acquisition Using Mobile Phones

Wearable sensors are small size mobile sensors designed to be worn on body. Most such wearable mobile sensors can be found in existing smartphones. Accelerometer, gyroscope,

ambient temperature sensor, light sensor, barometer, proximity sensor, and GPS are some of the sensing technologies that are built-in on most of the commercially available smartphones. Accelerometer sensors measure the acceleration of the device. The reading can be in one, two, or all three axes of X, Y, and Z. The raw data is represented as a set of vectors and returned together with a timestamp of the reading. Gyroscope is a sensor that measures the rotation rate of the device by detecting the roll, pitch, and yaw motions of the smartphone about the X, Y, and Z axes. Similar to accelerometer, readings are presented as time-stamped vectors. When the mobile device is attached to body involved in different activities, these two sensors generate different patterns in their transmitted signals.

5.4.2 *Data Preparation*

When collecting data for a long period of time, it is possible that the sensors temporarily freeze or fail to properly collect and store data for fractions of a second to a few seconds and in return, compensate for the missing data points by collecting data in a rate higher than the assigned frequency. In such cases, a preprocessing technique to fill in for missing data points and removing redundant ones can help insuring a continues and orderly dataset. Also, since the raw data are often collected with a high sampling rate, segmentation of the data helps in data compression and prepares data for feature extraction [146]. If segmentation is performed considering an overlap between adjacent windows, it reduces the error caused by the transition state noise [174]. The length of the window size depends on the sampling frequency and the nature of activities targeted for classification from which data is collected [174].

5.4.3 *Feature Extraction*

Feature is an attribute of the raw data that should be calculated [146]. In data analytics applications, statistical time- and frequency-domain features generated in each window are used as the input of the training process [104]. The ability to extract appropriate features depends on the application domain and can steer the process of retaining the relevant information. Most previous studies on activity recognition used almost the same features for training the models and classification of activities [175].

5.4.4 *Data Annotation*

Following data segmentation and feature extraction, the corresponding activity class labels should be assigned to each window. This serves as the ground truth for the learning algorithm and can be retrieved from the video recorded at the time of the experiment.

5.4.5 *Supervised Learning*

In supervised learning classification, class labels discussed in Subsection 5.4.4 are provided to the learning algorithms to generate a model or function that matches the input (i.e. features) to the output (i.e. activity classes) [104]. The goal is to infer a function using examples for which the class labels are known (i.e. training data). The performance of this function is evaluated by measuring the accuracy in predicting the class labels of unseen examples. Researchers have used different types of supervised classification methods for activity recognition [169, 171, 172].

5.4.6 *Model Assessment*

In order to determine the reliability of the trained model in detecting new examples of activity classes, part of the training dataset is used for testing the model. It is recommended that the test set is independent of the training set, meaning that the data that are used for testing have not been among the training data. For example, randomly chosen 10% of the training data can be left out so that the training is performed on the remaining 90% of data. Assessment of the model provides an opportunity for its fine-tuning so that certain variables (e.g. regularization factor to prevent over-fitting in neural networks) in the algorithm can be revised to yield the best possible model.

5.4.7 *Activity Recognition*

Once the model is trained and its parameters are finalized, it can be used for recognizing activities for which it has been trained. While data is being collected to determine the activities according to a trained classifier, such data can be stored in a dataset repository and be added to the existing training data, so that the model is further trained with a richer training dataset.

5.5 Experiments Setup and Data Analysis

Experiments were conducted in an outdoor workspace where different activities performed by multiple construction workers were imitated. These activities included sawing, hammering, turning a wrench, loading sections into wheelbarrows, pushing loaded wheelbarrows, dumping sections from wheelbarrows, and returning with empty wheelbarrows. Activities were performed in 3 different categories in order to assess certain circumstances (described in Subsection 5.2) in

the outcome of classification. A Commercially available armband was used to secure a smartphone on the upper arm of the dominant hand of each worker. Recent research on the selection of accelerometer location on bricklayer's body for activity recognition has shown that according to the body movement of the worker while performing different bricklaying activities, among 15 potential locations for wearing an accelerometer, the lower left arm and the upper right arm are the two best locations that yield the highest information gain [176]. In this study, the lower arm was not selected for recognition of the activities of interest since it precludes convenient execution of some activities. Consequently, the selection of the upper arm was expected to provide accurate and consistent results compared to other locations on the body. Figure 5.2 shows some snapshots of the construction workers wearing mobile phones on their upper arms while performing assigned activities in the experiments conducted in this research.

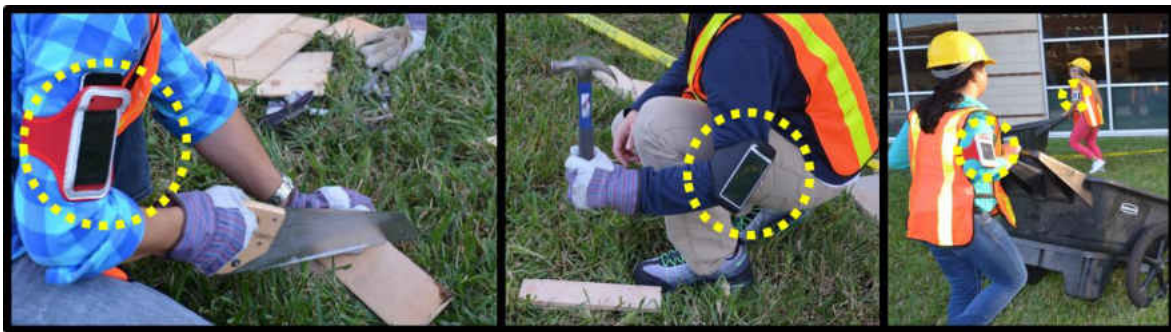


Figure 5.2: Data collection experiments (mobile devices are marked with dashed circles).

5.5.1 *Data Collection*

Smartphone built-in sensors and sensor logging applications in both Android and iOS operating systems were used for data collection. The sampling frequency was set at 100 Hz. This frequency is neither too low to miss any movement corresponding to the target activities, nor too

high to result in a large size for the collected data file. This sampling frequency has been also used in previous studies for accelerometer based activity recognition [177, 178, 179]. Data was collected in all 3 axes (X, Y, Z) from accelerometer and gyroscope. Construction workers were asked to do their assigned activities for a certain period of time while waiting for a few seconds in between each instance of their assigned activities. Each activity was performed by two subjects for later user-independent evaluations. Two subjects performed only sawing. In this case, the goal of activity recognition was to differentiate between the time they were *sawing* and the time they were *not sawing*. Two other subjects performed hammering and turning a wrench. In this case, the activity recognition was intended to detect the time they were *hammering*, the time they were *turning the wrench*, and the time there were not doing any of the two activities. Finally, the last two subjects were responsible for pushing the wheelbarrow and loading/unloading the sections. Therefore, the activities to be recognized in this case were *loading* sections into a wheelbarrow, *pushing* a loaded wheelbarrow, *dumping* sections from a wheelbarrow, and *returning* with an empty wheelbarrow.

Time-stamped data were logged into comma separated values (CSV) spreadsheets. The entire experiment was videotaped for data annotation. Time-stamped data from accelerometer and gyroscope were also synchronized and subsequently matched with the timer of the video during the data annotation process.

5.5.2 *Data Analysis*

Table 5.1 shows the number of data points collected per sensor per axis. Since classifications are conducted in 3 activity categories, the numbers of collected data points are tabulated and

reported for each category. Category 1 includes only one distinguishable activity, *sawing*, to assess the performance of the classifiers in detecting value-adding versus non-value-adding instances in a pool of accelerometer and gyroscope data. The result of classification in this category contributes to the overall performance of the developed activity recognition system when used for productivity measurement. In this category, *sawing* is categorized against *idling*. Category 2 includes instances of consecutive *hammering* and *turning a wrench* as two adjacent activities with almost similar corresponding movements of the worker’s arm. These two activities are also classified against *idling* to assess the accuracy of the developed activity recognition system in differentiating between activities that produce similar physical body motions. Finally, in category 3, four activities that produce different distinguishable body movements are categorized. These activities include *loading sections into a wheelbarrow*, *pushing a loaded wheelbarrow*, *dumping sections from a wheelbarrow*, and *returning an empty wheelbarrow*, that were also categorized against *idling*. Multiplication of the number of data points by 6 will result in all data points collected from two sensors in three axes.

Table 5.1: Collected data points per sensor per axis in each activity category.

Category	Activity	Number of Data Points per Sensor per axis
1	Sawing	120,755
2	Hammering + Turning a Wrench	149,682
3	Loading + Hauling + Unloading + Returning	337,800

In order to make up for the missing data and remove redundant data collected in a higher rate than the assigned sampling frequency, the timestamps of the adjacent collected data points were examined. Considering the 100 Hz sampling frequency, the normal difference between the two adjacent timestamps must be around 0.01 seconds. Therefore, in the data preparation phase, if this difference is greater than 0.015 seconds, the X, Y, and Z values of the missing data point were interpolated as the average of the two adjacent data points. As for the redundant collected data, any data point collected within less than 0.005 seconds of the last collected data point was removed. This assures the compatibility of the collected data with the recorded videotape for data annotation. As far as data segmentation was concerned, every 128 data points were segmented in one window and considering the 100 Hz sampling frequency, each window amounts to 1.28 seconds of data collection. The choice of 128 data points was due to conversion of the time domain to the frequency domain using fast Fourier transform (FFT) in which the window size should be of power of 2 [103, 104]. If the window size is not a power of 2, zeros will be added to the end of the window or it would be truncated to become a power of 2. With regard to the overlapping of the adjacent windows, previous studies for accelerometer-based activity recognition have suggested a 50% overlap between the adjacent windows [104, 180] and hence, a 50% overlap was also considered for data analysis in this research.

Moreover, common features used for activity recognition found in literature [175] were selected in this study and extracted from the raw data. In particular, mean, maximum, minimum, variance, root mean square (RMS), interquartile range (IQR) and correlation between each two pairs of axes comprised the seven time-domain features and spectral energy and entropy were the two frequency domain features. Considering data collection in three axes of the two sensors and nine

independent features extracted per sensor per axis, a total of 54 features were extracted from all collected data. Labeling windows was performed manually according to the recorded video of the data collection experiment.

5.5.3 Classifier Training

Five different classification techniques are used in order to systematically evaluate their performance in accurately detecting worker activities. In particular, neural network, decision tree, K-nearest neighbor (KNN), logistic regression, and support vector machine (SVM) are employed for classification. Decision tree, KNN, and SVM have been previously used for activity recognition [104, 177, 178] so they are selected in this study to evaluate their performance for classifying construction activities. However, neural network and logistic regression were examined to a much lesser extent [181].

5.5.3.1 Neural Network

The architecture of the neural network used for recognizing the activities is depicted in Figure 5.3.

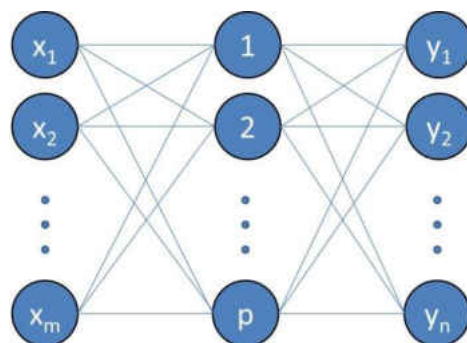


Figure 5.3: The architecture of the neural network used in this research.

The network follows a simple pattern of one input, one hidden, and one output layer. Considering the 54 features that serve as the input of the neural network, the input layer has $m=54$ units. The hidden layer consists of $p=25$ units. The number of units for the output layer is equal to the number of activity classes, n in each case. Considering the large feature space and in order to prevent overfitting, regularization was used. Using a regularization parameter, the magnitude of the model weights decreases, so that the model will not suffer from high variance to fail to generalize to the new unseen examples [182]. The activation function (i.e. hypothesis) used for minimizing the cost function in the training process is a Sigmoid function shown in Equation 5.1,

$$h_{\Phi}(x) = \frac{1}{1+e^{-\Phi x}} \quad (5.1)$$

in which $h(x)$ is the activation function (i.e. hypothesis), Φ is a matrix of model weights (i.e. parameters), and x is the features matrix. In this study, in order to minimize the cost function, the most commonly used neural network training method, namely feed-forward backpropagation is used. Considering a set of randomly chosen initial weights, the backpropagation algorithm calculates the error of the activation function in detecting the true classes and tries to minimize this error by taking subsequent partial derivatives of the cost function with respect to the model weights [183].

5.5.3.2 Decision Tree

Decision tree is one of the most powerful yet simplest algorithms for classification [39]. The decision tree method that is used in this research is classification and regression tree (CART). CART partitions the training examples in the feature space into rectangle regions (a.k.a. nodes)

and assigns each class to a region. The process begins with all classes spread over the feature space and examines all possible binary splits on every feature [39]. A split is selected if it has the best optimization criterion which is the Gini diversity index in this research, as shown in Equation 5.2,

$$I_G(f) = 1 - \sum_{i=1}^k f_i^2 \quad (5.2)$$

in which I_G is the Gini index, f_i is the fraction of items labeled with value i and k is the number of classes. The process of splitting is repeated iteratively for all nodes until they are *pure*. A node is considered *pure* if it contains only observations of one class, implying a Gini index of zero, or that there are fewer than 10 observations to split.

5.5.3.3 *K-Nearest Neighbor (KNN)*

Similar to the decision tree and unlike the neural network, KNN is a simple algorithm. Training examples identified by their labels are spread over the feature space. A new example is assigned to a class that is most common amongst its K nearest examples considering the Euclidean distance that is used as the metric in this research and as appears in Equation 5.3,

$$D = \sqrt{(x_i^{(1)} - x_{new}^{(1)})^2 + (x_i^{(2)} - x_{new}^{(2)})^2 + \dots + (x_i^{(d)} - x_{new}^{(d)})^2} \quad (5.3)$$

in which D is the Euclidean distance, x_i is an existing example data point which has the least distance with the new example, x_{new} is the new example to be classified, and d is the dimension of the feature space.

5.5.3.4 Logistic Regression

Logistic regression is a type of regression problems in which the output is discretized for classification [184]. Logistic regression seeks to form a hypothesis function that maps the input (i.e. training data) to the output (i.e. class labels) by estimating the conditional probability of an example belonging to class k given that the example actually belongs to the class k . This is accomplished by minimizing a cost function using a hypothesis function and correct classes to find the parameters of the mapping model [184]. The hypothesis function used in this research is the same as the activation function introduced in Equation 5.1 (the Sigmoid function) and thus the cost function to minimize is as shown in Equation 5.4,

$$J(\theta) = -\frac{1}{m} [\sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\theta}(x^{(i)}))] \quad (5.4)$$

in which $J(\theta)$ is the cost function, m is the number of training examples, $x^{(i)}$ is the i th training example, and $y^{(i)}$ is the corresponding correct label. Once the cost function is minimized using any mathematical method such as the Gradient Decent [184] and parameters are found, the hypothesis will be formed. In multi-class classification, the one-versus-all method is used to determine if a new example belongs to the class k [184]. Therefore, considering k classes, k hypothesis functions will be evaluated for each new example and the one that results in the maximum hypothesis is selected.

5.5.3.5 Support Vector Machine (SVM)

Compared to decision tree and KNN, SVM is considered as a more powerful classification algorithm. Although it has been widely used in vision-based pattern recognition and classification problems, some researchers [39] used it for classifying daily activities and thus its

performance is also assessed in this research. In a nutshell, SVM tries to maximize the margin around hyperplanes that separate different classes from each other. SVM can benefit from a maximum margin hyperplane in a transformed feature space using kernel function to create non-linear classifiers. The kernel function used for non-linear classification in this research is Gaussian radial basis function (rbf) which has been successfully applied in the past to activity recognition problems [104]. Further description of SVM models are out of the scope of this study but can be found in [39].

5.6 Results and Discussion

The performance of the classifiers is assessed in two ways. First, the training accuracy of each classifier was calculated. This means that all collected data points were used for both training and testing which provided an overall insight into the performance of a host of classification algorithms in recognizing construction worker activities using accelerometer and gyroscope data. Next, a more robust approach in evaluation of classifiers was adopted. In particular, 10-fold stratified cross validation was used and the results of the 10 replications of the training and testing were averaged out to report the overall accuracy. In k -fold cross validation, data are divided into k parts with almost equal number of data points. Then, in k recursive steps, one part is left out for testing and the remaining $k-1$ parts are used for training. In “stratified” version of k -fold cross validation, the k fold segmentation is done in a way that the proportion of the data from every class in each of the k parts remains the same as that of the entire training data [184]

The classification accuracies are reported for 3 activity categories listed in Table 5.1. The following activity codes are used in reporting the results: in the first category, activity *sawing*

(*SW*) and *being idle (ID)* are classified. In the second category, activities *hammering (HM)*, *turning a wrench (TW)*, and *being idle (ID)* are classified. Finally, in the third category classification is performed on the activities *loading sections into wheelbarrow (LW)*, *pushing a loaded wheelbarrow (PW)*, *dumping sections from wheelbarrow (DS)*, *returning an empty wheelbarrow (RW)*, and *being idle (ID)*. Table 5.2 shows the results of training and 10-fold cross validation classification accuracy of both subjects performing activities of category 1.

Table 5.2: Classification accuracy (%) for category 1 activities.

Category1		Neural Network	Decision Tree	KNN	Logistic Regression	SVM
Training	Subject I	100.00	99.36	98.08	98.72	98.19
	Subject II	99.25	99.15	97.34	98.08	97.33
10-fold CV	Subject I	96.77	96.06	95.95	96.05	96.91
	Subject II	97.02	95.42	96.27	96.70	96.59

According to Table 5.2, over 99% training accuracy was achieved for both subjects in category 1 using neural network classifier. This confirms the hypothesis that IMU data pertaining to a single activity performed by different workers contain highly distinguishable patterns. However, training accuracy is not an appropriate measure to assess the ability of using such data for new instances of the same activity. Nevertheless, the stratified 10-fold cross validation results confirm that regardless of the nature of classification algorithm, a single activity can be recognized with over 96% accuracy using all five classifiers. A thorough exploration of classification results within each category can help understanding the accuracies of each one of the activities versus the non-value-adding (i.e. *idling*) state. To achieve this, the confusion matrices of 10-fold stratified activity classifications for both subjects resulted from the best classifier (i.e. neural

network) are shown in Figure 5.4. Given that it is very likely that a construction worker performs more than one highly distinguishable activity at a time, activities performed in category 2 are designed such that they produce almost the same physical arm movement. Table 5.3 shows the training and 10-fold cross validation classification accuracy results of both subjects performing activities of category 2.

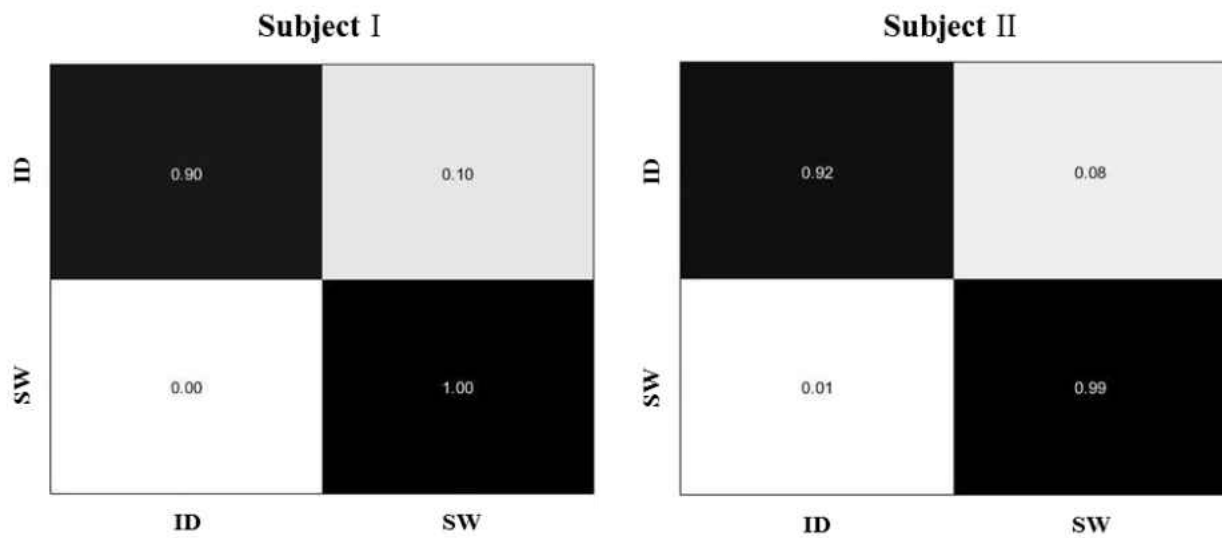


Figure 5.4: Confusion matrices of 10-fold cross validation of neural network classification for category 1 activities.

Table 5.3: Classification accuracy (%) for category 2 activities.

Category2		Neural Network	Decision Tree	KNN	Logistic Regression	SVM
Training	Subject I	98.62	97.07	93.81	88.14	87.28
	Subject II	93.30	94.67	91.67	84.03	83.43
10-fold CV	Subject I	93.19	85.83	87.80	86.42	85.34
	Subject II	86.64	78.20	83.35	81.02	81.72

Similar to category 1, the training accuracies are high particularly for the neural network classifier and the decision tree. While CART decision trees are not very stable and a small change in the training data can change the result drastically as appears in the outcome of the 10-fold cross validation, neural network presents an average of around 90% accuracy for both subjects. This is while all other classification methods performed almost the same with a slight superiority of KNN relative to the other algorithms. This result is particularly important considering the fact that the two activities in category 2 (i.e. *hammering* and *turning a wrench*) produce almost similar physical movements in a worker's arm. Figure 5.5 shows how these two activities are classified using 10-fold cross validation of the result obtained from neural network.

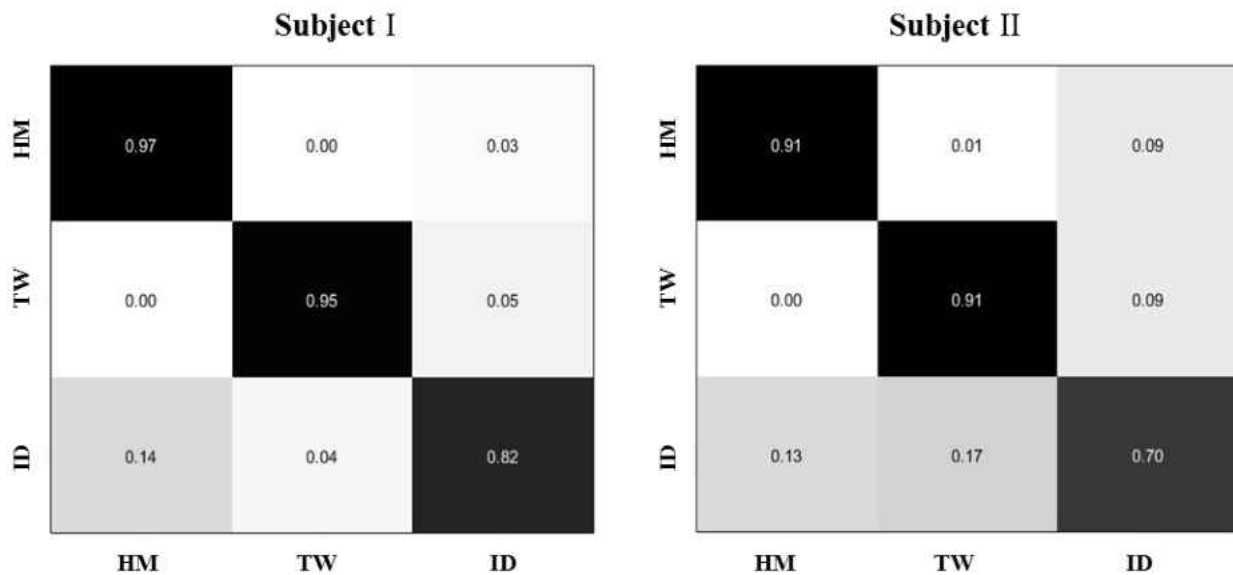


Figure 5.5: Confusion matrices of 10-fold cross validation of neural network classification for category 2 activities.

As appeared in Figure 5.5, both activities have been in fact classified with a high accuracy and the major contributor to lowering the overall accuracy was the idling state. This can be justified by the fact that the non-value-adding state may include different forms of physical movements in

case different activities are performed. In other words, the ID class includes various movements of different types so that relative to other two activities, more instances have been misclassified.

In the third category, a mixture of different distinguishable activities performed by typical construction workers is included to evaluate the performance of the developed activity recognition system in recognizing them. Table 5.4 shows the training and 10-fold cross validation classification accuracy results of both subjects performing activities of category 3.

Table 5.4: Classification accuracy (%) for category 3 activities.

Category3		Neural Network	Decision Tree	KNN	Logistic Regression	SVM
Training	Subject I	94.80	97.11	95.75	90.37	85.82
	Subject II	90.37	96.58	94.96	87.83	79.12
10-fold CV	Subject I	92.01	87.95	90.75	90.75	84.42
	Subject II	88.90	87.12	86.74	86.51	78.55

According to Table 5.4, again decision tree gained a high accuracy in training while as expected; its performance is not the same in 10-fold cross validation evaluation. However, except for the decision tree and SVM, all other classifiers, namely neural network, KNN, and logistic regression resulted in around 90% average accuracy for both subjects. Similar to the other two categories, the feedforward back-propagation implementation of the neural network resulted in the highest accuracy among all. Figure 5.6 shows how different activities in this category are classified using 10-fold cross validation of the result obtained from neural network.

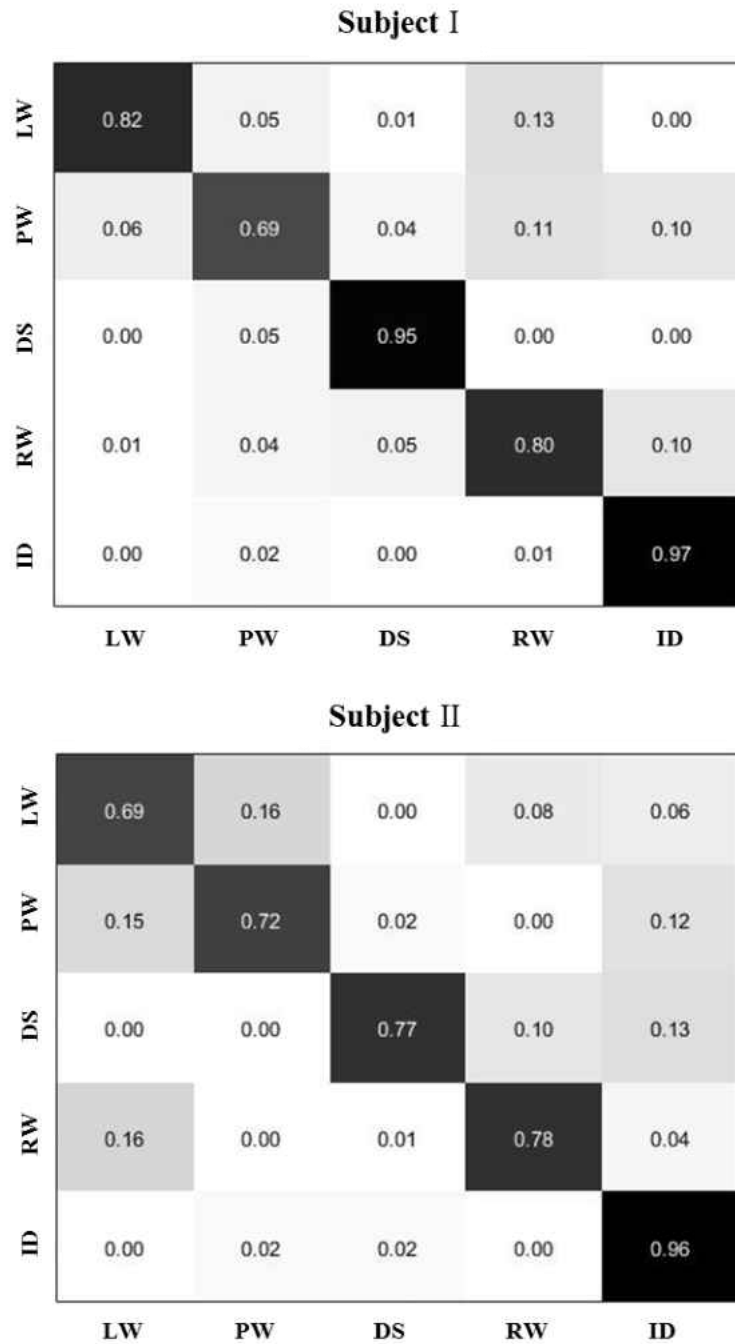


Figure 5.6: Confusion matrices of 10-fold cross validation of neural network classification for category 3 activities.

Based on the confusion matrices of Figure 5.6, the non-value-adding or *idling* state was classified properly in both cases. The most confused activities are LW and RW, particularly for the first subject, and LW, PW, and RW for the second subject. This might be due to the fact that LW and RW result in similar body movement patterns, while as confirmed in the two presented cases, different humans perform various activities with slightly different body movements (function of body height, body shape, ...). This may result in some confusion between two or more activities in each case.

After classifying the activities of each category based on the individual data received from each subject, the data collected from both subjects were combined to perform another round of classification. This evaluation allows further investigation of whether appending new data collected in future instances to existing data warehouse would result in acceptable classification and recognition of activities. Table 5.5 shows the result of the classification of combined data in all three categories.

Table 5.5: Classification accuracy (%) for combined data of subjects I and II in all three activity categories.

Combined Data for Subjects I & II		Neural Network	Decision Tree	KNN	Logistic Regression	SVM
Training	Category 1	99.75	99.04	97.71	97.39	97.23
	Category 2	91.67	95.49	92.27	82.86	82.86
	Category 3	89.49	96.48	92.46	86.41	78.62
10-fold CV	Category 1	96.27	95.58	96.22	96.54	96.64
	Category 2	87.78	78.57	87.73	82.23	82.18
	Category 3	88.17	85.62	87.68	85.84	78.34

According to Table 5.5, all categories have training accuracies of more than 90% in at least one classification algorithm. This promising result indicates that there exist classifiers that can categorize activities of different natures using combined data collected from wearable IMUs in different instances. In case of new examples, considering the robust 10-fold cross validation technique, while logistic regression's and to a larger extent, KNN's performance is very close to that of neural network, again neural network outperforms all the other classifiers. Figure 5.7 shows the confusion matrices of neural network for combined data of all three categories.

The last evaluation of classifiers' performances is conducted for the case of using data from one subject as the training set to classify activities of the second subject. This assessment is particularly important when trained model with the existing data is sought to be used for newly collected data. Table 5.6 shows the results of training each classifier using the data collected from subject I/II and tested on the data collected from subject II/I. In each category, the "I on II" row indicates that the classifiers were trained using the subject I data and tested on subject II data, and the "II on I" row indicates that the classifiers were trained using the subject II data and tested on subject I data.

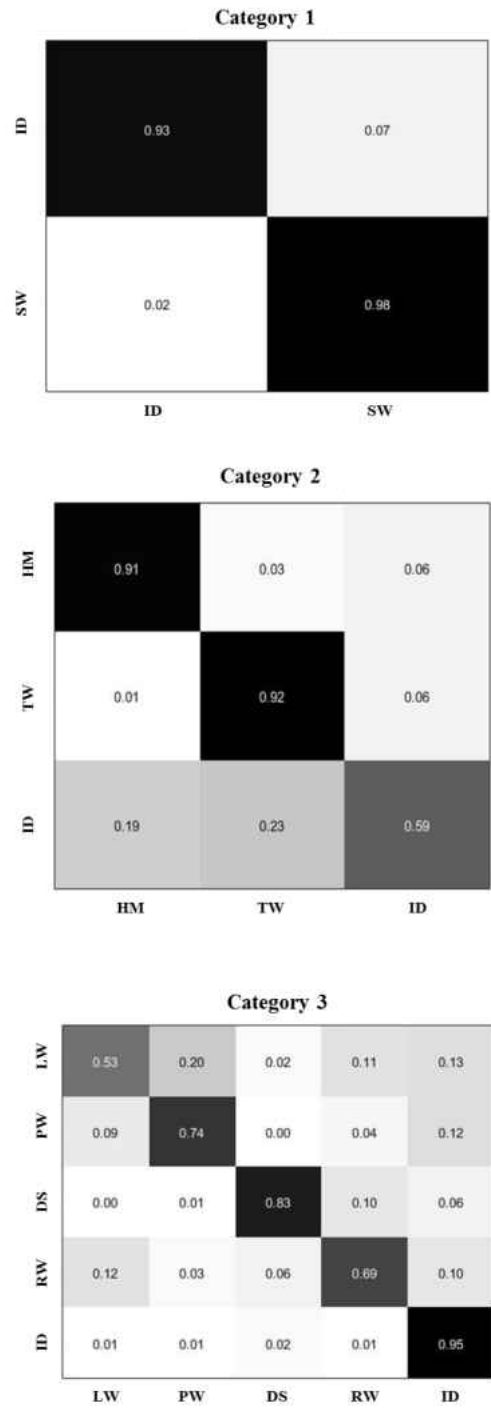


Figure 5.7: Confusion matrices of 10-fold cross validation of neural network classification for combined data of subjects I and II in all three activity categories.

Table 5.6: Accuracy (%) of classifiers trained with data from one subject and tested on data from another subject.

		Neural Network	Decision Tree	KNN	Logistic Regression	SVM
Category 1	I on II	94.24	94.78	96.05	93.71	94.04
	II on I	95.73	92.00	96.42	96.26	93.07
Category 2	I on II	62.10	63.05	68.20	64.93	63.30
	II on I	73.65	55.10	78.30	80.53	80.93
Category 3	I on II	78.85	73.66	79.23	76.62	72.45
	II on I	77.86	57.31	79.79	78.92	71.49

Comparing different classifiers, it is apparent from Table 5.6 that KNN has the best classification accuracy which is even slightly better than neural network in this case. This is true for all the categories and thus indicates the power of KNN (despite its simplicity) in generalizing a trained model to new examples. Comparing different activity categories in this scenario, while classification of category 1 activities with only one distinguishable activity results in an accuracy of more than 96%, classification of activities in the other two categories have resulted in less accuracies. In particular, category 2 with two similar activities shows a less accurate performance. Nevertheless, while category 3 classification was performed using 5 different classes, an accuracy of around 80% in the best case (i.e. KNN) shows promising results when a rich data warehouse is available.

5.7 Summary and Conclusions

In spite of its importance, automated recognition of construction worker activities on the jobsite has not been given due attention in CEM literature. While some efforts have been made in the

past to develop vision-based techniques for automated tracking and recognition of construction entities, the state-of-the-art in employing IMU sensors with wide variety of applications in other domains has not been yet explored within the CEM context. The presented Chapter discusses a novel methodology for designing and testing a low-cost pervasive construction worker activity recognition system capable of detecting activities of various natures that are typical to construction jobsites. Towards this goal, built-in sensors of ubiquitous smartphones have been employed to assess the potential of wearable systems for activity recognition. Smartphones were affixed to workers' arms using sport armbands, and accelerometer and gyroscope data were collected from multiple construction workers involved in different types of activities.

The high levels of training accuracies achieved by testing several classification algorithms including neural network, decision tree, K-nearest neighbor (KNN), logistic regression, and support vector machine (SVM) confirmed the hypothesis that different classification algorithms can detect patterns that exist within signals produced by IMUs while different construction tasks are performed. Through 10-fold stratified cross validation, algorithms were trained with 90% of the available data and the trained models were tested on the remaining 10%. In different categories of activities, around and over 90% accuracy was achieved. This promising result indicates that built-in smartphone sensors have high potential to be used as integrated data collection and activity recognition platforms in construction environments.

Further investigations were made by combining the data from different subjects. In the first two categories with less activities to be classified, accuracies of more than 90% were achieved which indicate that combination of data collected from different workers can result in promising

outcome for activity recognition. When the number of activities increased and more similar activities were sought to be classified (i.e. category 3) the recognition accuracy fell to 70%-80%. In the last assessment, data from each subject were used to train two different classifiers. The trained models were then tested using the data collected from another subject. While this scenario introduced the most challenging situation, KNN was able to present around 95%, 75%, and 80% accuracies. It is worth mentioning that in terms of computational time, KNN is highly superior to neural network as it is much less complex because there is no need for an optimization process with high iteration numbers. KNN simply compares the test data to the training data and that is why it is also referred to as a “lazy learner” [39].

Overall, results indicated that the CEM domain similar to other sectors such as health and fitness, medicine, and elderly care can benefit from the applications of activity recognition on construction jobsites. Some application areas include productivity measurement, progress evaluation, labor training programs, and safety and health management.

5.8 Future Work

A potential direction for future work in this area will be to explore whether the results achieved so far can be used for automatically extracting process knowledge such as activity durations and precedence logic for the purpose of ubiquitously updating and maintaining simulation models corresponding to field operations. In addition, another branch of future work rooted in the current research is automated identification of unsafe workers’ postures in physically demanding construction activities. Work-related Musculoskeletal Disorder (WMSD), back, knee, and shoulders injuries are among the most common injuries that can be prevented or reduced by

complying with Occupational Safety and Health Administration (OSHA) or the National Institute for Occupational Safety and Health (NIOSH) standards and rules [185, 186].

Productivity measurement and improvement is another direction for future work of this study. There has been a great deal of research on different techniques for productivity evaluation, tracking, and improvement in construction industry such as construction industry institute (CII) productivity measurement methods [187], the construction productivity metric system (CPMS) [188], activity/work sampling [189, 190], and recent studies targeting the relationship between task-level productivity and physical movements such as the study conducted by Gatti et al. [191]. In particular, it is possible to calculate the proportion of time dedicated by each worker to each activity. For example, Figure 5.8 shows pie charts indicating the proportions of time dedicated to each activity in the experiments conducted in this research, as discovered by the designed activity recognition system.

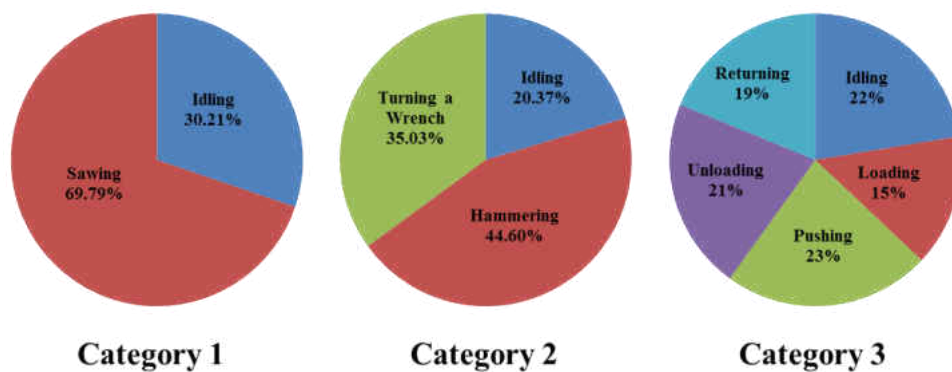


Figure 5.8: Discovered time allocation proportions in the conducted experiments, for productivity measurement.

The discovered knowledge presented in this Figure is of great importance to the process of productivity measurement and improvement. Particular to the activity/work sampling, this

information can help automate the process, thus significantly reducing the manpower required for manual analysis and potential errors and inconsistencies associated with manual observations.

CHAPTER 6: DATA-DRIVEN SIMULATION OF CONSTRUCTION PROCESSES WITH COMPLEX WORKER INTERACTIONS USING SMARTPHONE SENSOR-BASED ACTIVITY RECOGNITION

6.1 Introduction

In order to systematically evaluate the performance and added value of the activity recognition techniques described in Chapters 4 and 5 to data-driven simulation modeling of construction and infrastructure projects, in this Chapter, a relatively complex operation consisting of multiple crews and several processes is modeled and described. Activities in this operation are carried out by workers who interactively contribute to each process and to the overall goal of the operation which is to install prepared and transported wooden sections. To have a true benchmark for obtaining real-world process-level data as well as verifying and validating the simulation results, the operation was replicated in an experimental setting by workers each wearing an armband that carried a smartphone using which accelerometer and gyroscope data were collected in real time.

Following the completion of the operation and data collection, the entire operation was modeled using discrete event simulation (DES). The choice of DES for this purpose was due to the fact that most construction and infrastructure projects consist of discrete activities or sub-systems which make them ideal for DES modeling. In particular, two separate DES models were created; in model 1 (data-driven model) activity durations were extracted from sensory data using the activity recognition framework described in Chapter 5, whereas in model 2 (static model) activity durations were estimated based on certain heuristics (e.g. instructions given to workers during the experiment, physical dimensions of the workspace, approximated movement speeds).

Since in stochastic simulations, activity durations are added to the model in form of probability distributions, after extracting durations of each instance of an activity, probability distributions are fit to the duration values and used to describe that activity in the simulation model.

The rest of this Chapter will first discuss the experiment setup and description of the activities involved. Next, the modeling platform and simulation design of the operation are described. The activity recognition framework and its application in this scenario are then investigated and extracted activity durations will be presented. Finally, results of static and data-driven simulation models are compared and conclusions are made.

6.2 The Operation Experiment Design

6.2.1 *Experiment Setup*

The goal of the operation that was replicated in an experimental setting was to prepare, transport, and install wood sections. The experiment was conducted in an outdoor environment that resembled a small construction jobsite. Figure 6.1 shows a schematic illustration of the operation. As shown in this Figure, the cyclic operation starts with a worker, *W1*, who saws lumber inside an imaginary *wood workshop* and prepare wood sections of proper sizes and shapes. These sections are then transported to the *installation area* by two other workers, *W2* and *W3* who are tasked with loading the sections, hauling them to the installation area, and dumping them where an installer worker, *W4*, is waiting to receive the sections and install them in their positions. Figure 6.2 is a real snapshot of the experiment.

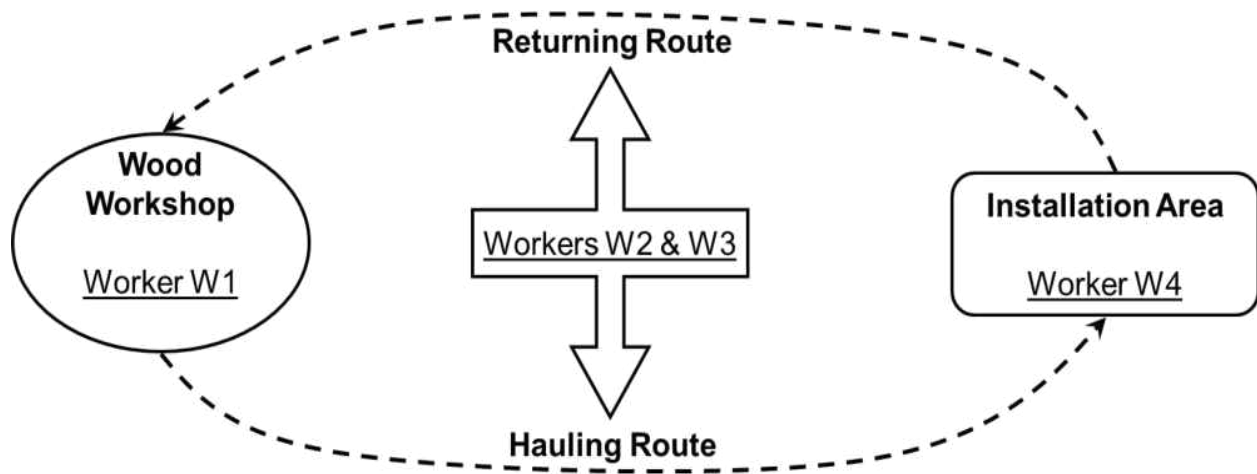


Figure 6.1: Schematic illustration of the operation experiment setup.



Figure 6.2: Snapshot of the operation showing four workers performing the experiment.

Each process involves one or more activities assigned to different workers. In the *wood workshop*, the process of cutting the lumber pieces consists of only one activity, *sawing*, carried out by worker *W1*. The transportation process involves four activities namely putting sections into the wheelbarrow or *loading*, pushing a loaded wheelbarrow or *hauling*, dumping the sections in the installation area or *unloading*, and returning the empty wheelbarrow or *returning*. Workers *W2* and *W3* are responsible for the transportation process. Finally, worker *W4* is tasked

with the installation process which involves the activities *hammering* and *turning the wrench*. All the aforementioned processes, activities, and tasked workers are summarized in Table 6.1.

Table 6.1: List of the processes involved in the operation and activities within each process.

Process	Activity	Worker
Cutting Lumber	Sawing	W1
Transportation	Loading	W2 & W3
	Hauling	
	Unloading	
	Returning	
Installation	Hammering	W4
	Turning the Wrench	

6.2.2 *Assumptions and Rules*

The *Loading* and *unloading* activities follow underlying operational rules that are enforced in the experiment and later in the simulation models. The first operational rule is applied to the *loading* activity; in particular *loading* will not be executed until there are at least two wood sections available for transportation. Therefore, when there are less than two sections prepared by worker *W1*, and either or both workers *W2* and *W3* are available, they will wait in a queue until at least two sections are ready for loading. With the same token, if either or both workers *W2* and *W3* are available, one section should wait until there is at least one more section prepared by *W1* so that both sections can be loaded. The second operational rule is for *unloading*. It is assumed that the space available for unloaded sections is enough only for two sections and *unloading* activity should be executed in only one instance, meaning that if there is any section waiting to be processed by worker *W4*, the available workers *W2* or *W3* should wait until there is no section

awaiting installation process. The last operational rule that is applied to both *loading* and *unloading* activities is that only one instance of each activity can be performed at any given time, meaning that simultaneous execution of either *loading* or *unloading* activity is not allowed.

6.3 Simulation Model of the Operation

The operation described in Section 6.2 was carefully modeled in Stroboscope (STate and ResOurce Based Simulation of CONstruction ProcEsses), a DES scripting environment based on Activity Cycle Diagrams (ACDs) that is designed for the simulation of processes common to construction engineering [51]. Simulation models created in Stroboscope are based on a network of interconnected modeling elements described in a script containing programming statements that give the elements unique behavior and control the simulation [34]. This network of the interconnected elements (a.k.a. the ACD) is designed to be similar in appearance and function to CYCLONE simulation platform, which was the first system developed specifically for construction operations [129]. The ACD of the operation described in Section 6.2 is shown in Figure 6.3.

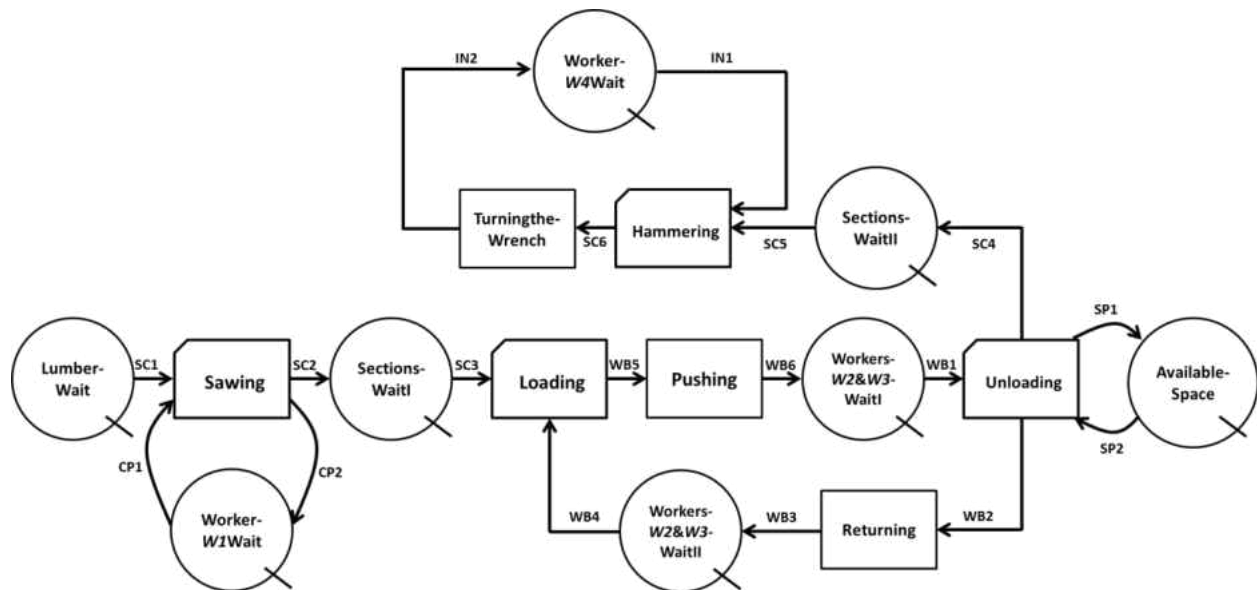


Figure 6.3: The ACD of the operation for modeling in Stroboscope.

In Figure 6.3, resources move from each node to the succeeding node in the direction shown by the connection link. A circle with a slash in the bottom right corner is a *Queue* that serves as the storage location for the resources. A rectangle with a cut-off in the top-left corner is called a *Combi* and a regular rectangle is called a *Normal*. These two nodes represent two different types of activities and hold the resources for the amount of time determined by activity durations. In particular, a *Combi* is always preceded by a *Queue* while a *Normal* activity cannot be preceded by a *Queue*. In Figure 6.3, `LumbersWait` holds lumber pieces before they are taken by worker `W1` for activity `Sawing`. The `WorkerW1Wait` Queue populated with 1 entity (i.e. 1 worker) ensures that only one instance of the `Sawing` activity is carried out in any point of time. Upon being sawed, sections wait in `SectionsWaitI` Queue to be loaded for transportation. This Queue satisfies the first operational rule described in Subsection 6.2.2. The

WorkersW2&W3WaitII Queue is where Workers W2 and W3 are drawn from one by one to load only two sections, if available. Similar to SectionsWaitI, this Queue also contributes to satisfying the first operational rule. When enough sections and transportation workers are available, the Loading Combi is activated, lasts for its assigned duration, and then releases the captured resource (i.e. worker) to the Hauling Normal. Again, this activity will hold the resource for the amount of time determined by its corresponding duration. Next, according to the second operational rule in Subsection 6.2.2, Workers W2 and/or W3 wait in the WorkersW2&W3WaitI Queue before the space is available for activation of the Unloading Combi. Finally, the SectionsWaitII Queue is where at most two sections are being held before they can proceed to the Hammering Combi. It must be noted that the Hammering Combi will not be activated if either of the SectionsWaitII or WorkerW4Wait Queues does not have available resources. Such situation happens for example if worker W4 is captured by the TurningtheWrench Normal.

While the ACD shown in Figure 6.3 provides a high level representation of the simulated operation, more specific operational details are incorporated in the script of the model. This is where attributes of the queues and activities as well as the model parameters are assigned. Such attributes define how model parameters behave. For example, the followings show how some of the network elements inducing activities (Normal and Combi), Queues, and links are defined in Stroboscope:

```

/* Definition of Network Elements
QUEUE      SectionsWaitI  Sections;
COMBI      Loading;
NORMAL     Pushing;
QUEUE      WorkersW2&W3WaitI  Workers;
COMBI      Unloading;
LINK CP1   WorkerW1Wait   Sawing;
LINK CP2   Sawing         WorkerW1Wait;
LINK SC1   LumberWait     Sawing;
LINK SC2   Sawing         SectionsWait1;

```

Another key attribute is the durations of Combi and Normal activities. Activity durations are sampled from the specified probability distributions. In the next Subsection, the activity recognition framework developed in this research in order to extract realistic activity durations is described.

6.4 Duration Extraction through Activity Recognition

The activity duration extraction technique using smartphone sensors and machine learning classifiers was initially introduced in Chapter 4 for construction equipment. Human crew activity recognition using similar instruments and mechanism was also described in Chapter 5. In this Chapter, the results obtained in Chapter 5 for the classifier with the best performance are used to recognize activities within each process of the operation described. Once activities are

recognized and classified, activity durations are extracted and used inside the simulation model by means of probability distribution functions.

In Chapter 5, different machine learning classifiers were examined for construction worker activity recognition. The activities under examination were similar to those of interest in the operation that is simulated in this Chapter. Other configuration of the activity recognition framework such as the sampling frequency, data preparation procedures, segmentation window size, and extracted features are the same as the ones considered in Chapter 5. Table 6.2 presents a summary of this information.

Table 6.2: Sensory data collection configurations used for activity recognition.

Configurations	Mechanism or Values Used
Sampling Frequency	100 Hz for both accelerometer and gyroscope
Data Preparation	Interpolating missing data and removing data with close timestamp
Window Size	128 data points with 50% overlap
Extracted Features	Statistical time- and frequency domain using fast Fourier transform

It was concluded in Chapter 5 that out of the five classifiers of different natures, namely artificial neural network, decision tree, K-nearest neighbor (KNN), logistic regression, and support vector machine (SVM), artificial neural network had the best performance in recognizing and classifying all various activities. Therefore, artificial neural network is employed in this Chapter for recognition of the activities toward extracting their corresponding durations. However, in order to improve the results even further, an ensemble methodology is also adopted. In ensemble neural network, more than one network is trained using different subsets of the same training dataset. Subsequently, to classify a given input, all the trained networks will be used and the

classification decision will be determined through a consensus scheme [192]. Plurality voting is the scheme used in this research in which the class which receives the highest number of votes from different classifiers is the decision of the ensemble [193].

Bootstrap aggregation or Bagging is the ensemble algorithm used in this Chapter. In Bagging, T training data subsets each containing m training examples are selected randomly with replacement from the original training set of m examples. The classification result of the ensemble is determined through plurality voting [193]. Here, the number of training dataset is $T = 10$. Classification was performed on the activity level within each process, meaning that the result of the classification in terms of accuracy in correctly predicting the activities within each process is reported. It is worth mentioning that within each class, an extra activity is included as the *idling* state in which the worker is not contributing to any of the assigned activities within the process. Table 6.3 shows the Bagging ensemble result of classification accuracies for each process.

Table 6.3: Neural network Bagging classification accuracy (%) for recognizing activities within each process.

Process	Accuracy (%)
Cutting Lumber	99.25
Transportation	89.21
Installation	92.66

As seen in Table 6.3, the classification accuracy for the first process involving the *sawing* activity is almost perfect and it is expected that it closely matches the observed activity durations. This will be assessed in a more statistically vigorous way in the next Section.

However, activities that comprise the other two processes, namely transportation and installation have not been classified as accurately. Therefore, the durations extracted from these activities are expected not to be as close to the observed durations as the first process. However, it should be noted that the similarity of the extracted durations to the observed values does not necessarily conform the same accuracy as their associated activity recognition accuracy. In other words, although it is expected that the durations of activities within the cutting lumber process is predicted with the highest accuracy of all, the accuracy of predicting activity durations for the transportation and installation processes may not follow the same results in terms of relative accuracies. This is due to the fact that extracting activity durations follows a heuristic algorithm according to which many of the misclassified instances are ignored. In essence, the algorithm first replaces instances of any different classes that are appeared within a large number of detected instances of the same class. For example, few instances of class C_2 classified after many instances of class C_1 followed by other instances of class C_1 are considered as class C_1 . The exact numbers followed by this heuristic algorithm depends on the sampling frequency, window size, and rough approximation of the activity durations. Here with sampling frequency of 100 Hz, window sizes of 128 data points with 50% overlap that amounts to 0.64 seconds of data, any two instances of an activity that normally takes more than 20 seconds but are separated out to less than 12 seconds are merged. Such heuristics result in improved accuracy for activity duration extraction.

6.5 Simulation Input Modeling

In this Section, the process of input modeling of the operation simulation is described. Simulation input modeling includes fitting probability distribution functions to the activity durations and has a high impact on the accuracy of the model. First, observed activity durations using the recorded videotape of the experiment are compared to those extracted through the activity recognition system. This step serves to guarantee that extracted activity durations are not statistically significantly different from those that actually took place in the real experiment. In case there is a statistically significant difference between the two sets of duration values, then it cannot be expected from the data-driven simulation model to output values close to the actual ones observed in the experiment.

In order to compare observed and extracted activity durations, the *student t-test* is used to evaluate the null hypothesis of no considerable difference between the expected and sample distributions. More details of the t-test are provided in Chapter 3. Table 6.4 shows the result of the t-test for activities within each process. As shown in this Table, the null hypothesis for none of the activities was rejected through comparison of the observed and extracted activity durations with 5% significance level. This confirms that the two sets of activity durations are not statistically significantly different.

Table 6.4: Comparison of the observed and extracted activity durations using student t-test.

Process	Activity	Observed Duration (Seconds)		Extracted Duration (Seconds)		p-value	Null Hypothesis
		Mean	SD	Mean	SD		
Cutting Lumber	Sawing	27.95	6.50	27.97	6.57	0.78	Not rejected
Transportation	Loading	8.96	1.40	9.24	1.75	0.27	Not rejected
	Pushing	14.02	2.43	14.14	2.81	0.63	Not rejected
	Unloading	13.18	1.96	13.53	2.01	0.08	Not rejected
	Returning	11.33	2.14	11.39	2.29	0.78	Not rejected
Installation	Hammering	17.05	2.48	17.59	2.46	0.09	Not rejected
	Turning the Wrench	13.39	3.42	13.44	3.35	0.75	Not rejected

The objective of creating simulation models of the operation experiment is to compare the results of the simulation created based on the extracted activity durations (data-driven model) to the one created according to the estimated activity durations (static model). To this end, estimated activity durations were defined by taking into account the [minimum, maximum], or three-point estimation [minimum, mode, maximum] durations for each activity which is a common practice in creating construction simulation models or project management schedules using project evaluation and review technique (PERT) [60]. These two schemes are in essence equivalent to sampling from uniform and triangular distributions. Therefore, these two probability distributions were considered for activity durations inside the static model. The parameters of the two probability distributions however were estimated according to two heuristics; the instructions given to the workers performing the activities, and engineering assumptions of the variance for such durations considering the nature of each activity. For example, worker *WI* was

asked to saw each piece of lumber for about 25 to 30 seconds. Therefore, the probability distribution considered for this activity was a uniform distribution with a minimum of 22 and maximum of 33 to account for 3 seconds of variations from the extrema. Table 6.5 shows the probability distributions fitted to the extracted activity durations along with those estimated for each activity.

Table 6.5: Probability distributions used inside the two simulation models.

Activity	Probability Distributions Used for	
	Extracted Duration (Data-Driven Model)	Estimated Durations (Static Model)
Sawing	Triangular[12,31.6,40]	Uniform[22, 33]
Loading	Triangular[6,8.1,13]	Uniform[5, 7]
Hauling	9 + Gamma[1.63, 3.16]	Uniform[8, 12]
Unloading	9 + 8 × Beta[1.83, 1.41]	Uniform[7, 12]
Returning	6 + Gamma[1.26, 4.28]	Uniform[7, 10]
Hammering	Normal[17.8,2]	Triangular[13,15,17]
Turning the Wrench	7 + 14 × Beta[1.62, 1.78]	Triangular[13,15,17]

The following sample lines show how extracted activity durations are defined inside Stroboscope:

```
DURATION Sawing 'Triangular[12, 31.6, 40]';
DURATION Loading 'Triangular[6, 8.1, 13]';
DURATION Pushing '9 + Gamma[1.63, 3.16]';
DURATION Unloading '9 + 8 * Beta[1.83, 1.41]';
DURATION Returning '6 + Gamma[1.26, 4.28]';
DURATION Hammering 'Normal[17.8, 2]';
```

```
DURATION TurningtheWrench '7 + 14 * Beta[1.62, 1.78]';
```

6.6 Performance of the Data-Driven Model vs. Static Simulation Model

Using the two sets of probability distributions shown in Table 6.5, two identical simulation models are created based on the ACD introduced in Section 6.3. The only difference between the two simulation models is in the activity durations defined in the input script of each model. Each model was run for 50 replications by generating random numbers from the same seed, and five measures were collected for comparison of the outputs of the simulations to the real world observations. The measures include the average waiting times (in seconds) of the entities in the four Queues namely `SectionsWaitI`, `SectionsWaitII`, `WorkersW2&W3WaitI`, and `WorkersW2&W3WaitII`, as well as the total operation time (in minutes). Figure 6.4 shows the comparison between the five measures. For each measure shown in this Figure, the first bar from the top refers to the value of that measure observed in the real-world operation. The second bar with a slightly lighter color corresponds to the mean of the average waiting times resulted from the data-driven simulation after 50 replications. The error bar refers to the standard deviation of these 50 replications. The third bar with the lightest color is the result of the static simulation created based on the estimated activity durations.

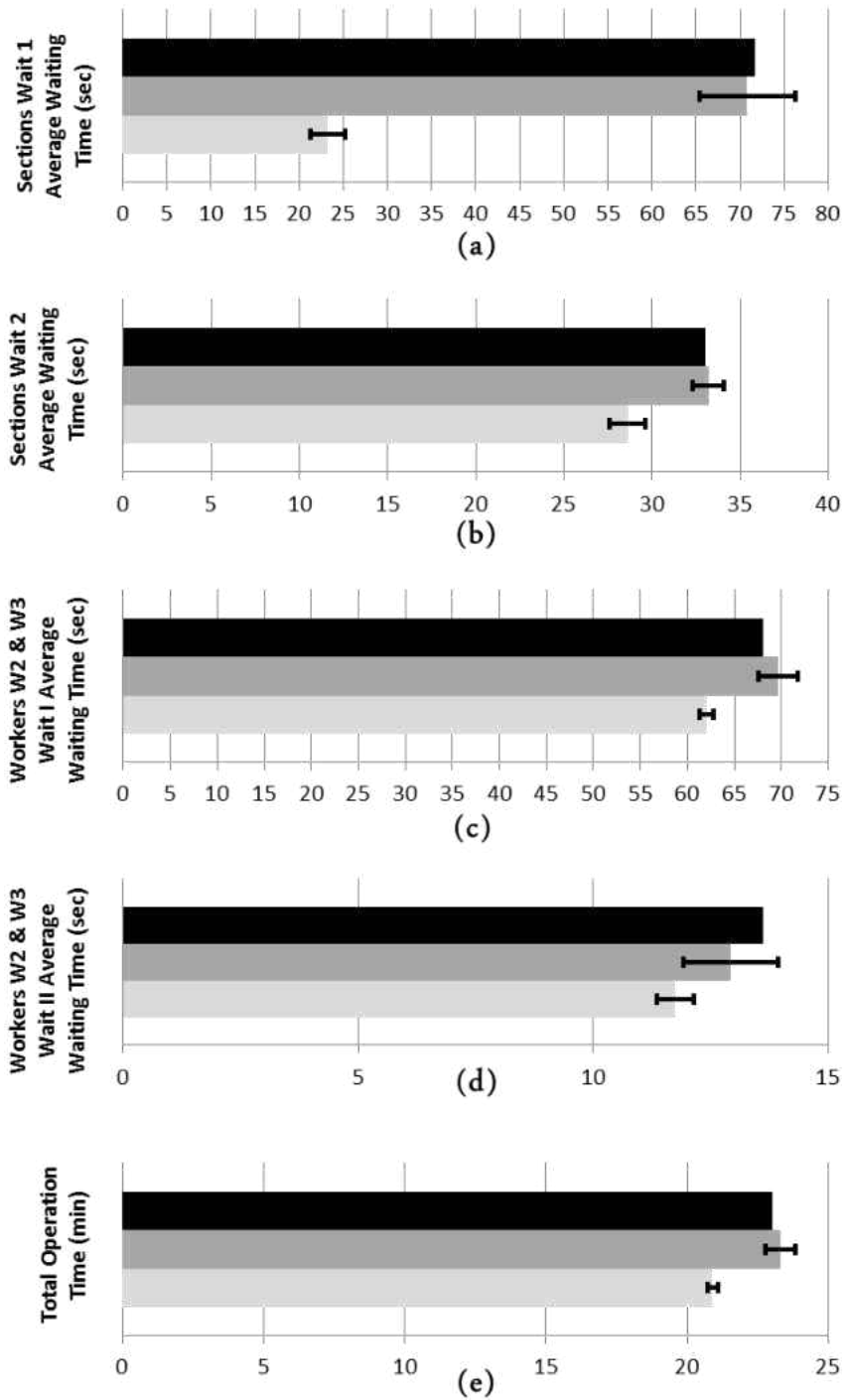


Figure 6.4: Comparison of results obtained from the real-world experiment, and the output of static and data-driven simulation models.

6.7 Discussion of the Results

According to Figure 6.4, the observed values for all five measures are within one standard deviation of the results obtained from the data-driven simulation model. This is while all the output measures obtained from the static simulation with estimated values are underestimating the waiting times and total duration of the operation. In fact, this is what happens most of the time in construction projects where simulation models created in the planning and pre-construction stage estimated significantly underestimate or overestimate the durations of the real world processes [60]. This is while uniform and triangular probability distributions (and not fixed values) were used for estimating activity durations inside the static model. It must be noted, however, that the underestimation observed in the output of the static simulation model in all five measures is particular to this specific example and cannot be generalized to other problems. More specifically, the measures obtained from the static model could have as well resulted in an overestimation. What is of utmost importance in interpreting the results is the noticeable difference between the outputs of the two simulation models and the fact that the result of the data-driven model is closer to real-world observations.

A considerable discrepancy can be seen in the result obtained from the static simulation and the observed value for the average waiting time in `Queue SectionsWait1`. This can be explained as follows; since the `WorkersW2&W3WaitII` average waiting times in bar chart (d) are very close (considering the scale of this chart), the availability of workers should not have influenced the difference. Therefore, it can be explained through the difference in durations considered for the `Sawing` activity. It turns out that the data-driven simulation with the probability distribution

of Triangular [12,31.6,40] for `Sawing` activity, samples from a lower range of numbers starting from 12 seconds, while the minimum value for the uniform distribution of the observed values is 23 seconds. This results in a much faster `Sawing` in reality which in turn provides more sections waiting in the `SectionsWait1Queue`. Other than `Sawing`, most of the other activities have estimated distributions resulting in sampling of lower values that explains the low overall operation time obtained from the static simulation.

Regardless of the reasons for any discrepancy between the extracted and estimated activity durations, the very fact that any difference in activity durations can substantially change the simulation output statistics verifies the significance of having more realistic simulation models through data-driven input modeling.

6.8 Summary and Conclusions

In this Chapter, a complex operation involving multiple interactions between workers performing construction activities was described and modeled in DES using process-level data collected from the crew in real time. Sensory data consisted of accelerometer and gyroscope data and were collected using smartphones affixed on workers' upper arms. Activities performed by the workers were then recognized and classified using the construction worker activity recognition methodology introduced in Chapter 5. Following activity recognition, corresponding activity durations were extracted and probability distributions were fit to the extracted durations. Moreover, these durations were compared to the values observed in the real world experiment to confirm their fidelity. Extracted activity durations were then fused into a data-driven DES model

created based on the experiment design in order to compare the results against those of a similar but static simulation model with estimated values for activity durations.

Analysis of the output obtained from the two simulation models with respect to five quantifiable measures (i.e. average waiting times of the entities in four Queues namely SectionsWaitI, SectionsWaitII, WorkersW2&W3WaitI, and WorkersW2&W3WaitII, as well as the total operation time) revealed that the data-driven simulation model created based on the knowledge (i.e. activity durations) extracted by the developed activity recognition framework outperforms the static simulation model created based on estimated activity durations. Considering the fact that often times the common practice in creating construction simulation models is using historical (secondary) information and subjective assumptions in designing model attributes, obtaining results in close agreement with reality reaffirms the significance of substituting this traditional approach in creating simulation models with a more robust and reliable data-driven and knowledge-based methodology that was described in this Chapter.

CHAPTER 7: CONCLUSIONS

Most heavy construction and infrastructure projects involve dynamic operations; resources (equipment, workers, and materials) are often in complex interactions with each other in unstructured arrangements within the site layout. Moreover, the processes are prone to uncertainties due to internal (e.g. equipment breakdown, work breaks, material shortage) or external factors (e.g. adverse weather, soil stability).

In such a highly dynamic and complex environment, real world project performance evaluation is often difficult (if not impossible), expensive, and time consuming. Therefore, computer simulation models have emerged to help decision-makers obtain a thorough understanding of the project progress and future performance in various fronts. In particular, discrete event simulation (DES) has been widely used in planning and scheduling of construction projects due to the fact that most construction operations can be broken down into discrete processes.

However, unrealistic outputs of the traditional construction simulation modeling approaches (due to their intrinsic limitations in design and implementation) have resulted in reluctance by most industry practitioners to deploying simulation as a value-adding decision support tool. The current practice tends to use static models in which estimations or engineering judgements made by subject matter experts serve as the basis for simulation input modeling and parameter estimation. Therefore, the current practice falls short in properly incorporating execution-phase data and precisely modeling specific conditions and complexities on the ground.

This long-standing limitation in construction simulation modeling and the ability to overcome this challenge have in fact motivated the presented doctoral research. This dissertation documented the work that led to the design and implementation of a novel approach that draws knowledge and key concepts such as process mining from the field of business process management, coupled with machine-learning-based activity recognition from the computer science domain in order to create data-driven knowledge-based construction simulation models. In the developed framework, multi-modal process data collected from field agents are analyzed to extract relevant and required knowledge for precise simulation modeling.

In Chapter 2, the feasibility, applicability, and functionality of data-driven construction simulation models were studied. As a motivating case, multi-modal data (i.e. position, orientation, and payload) was collected and processed using a rule-based taxonomy of construction field agents' states. Moreover, the approximate work zone vicinities of key areas in a jobsite were determined thorough k-means clustering.

The primary contribution of the research presented in Chapter 2 was developing a rule-based taxonomy of activities and states based on the collected data that enables knowledge extraction and results in superior data-driven simulation models that can outperform equivalent static models.

In Chapter 3, a thorough study was conducted on the algorithms that enable extracting knowledge pertaining to queue properties. The significance of focusing on queue formation is twofold in managing construction projects; first formation of the queues is inevitable in construction jobsites and it is necessary to carefully design and manage queues to prevent

bottlenecks in process flows and supply chain systems. Second, simulation models are the main tools for studying the dynamics and uncertainties involved in queuing systems and thus for developing a data-driven simulation model, special attention should be drawn to queue formations. In this Chapter, in particular, collected agent data was used to find interarrival times, service times, and queue discipline.

The primary contribution of the research presented in Chapter 3 was designing algorithms that enabled extracting and tracked changes in the properties of queues in construction jobsites.

In Chapter 4, a robust methodology was designed and implemented that employs built-in sensors of smartphones to recognize construction equipment activities. It was observed that the use of a wireless sensor network (WSN) such as the one introduced earlier in Chapter 2 may not be practical as it may introduce some implementation challenges in a real-world construction jobsite. Such challenges include but are not limited to maintenance and calibration problems due to ambient factors such as dust and rain, prohibitive cost especially when dealing with a large construction fleet, and data synchronization, storage and transmission issues. To remedy these potential problems, in this Chapter, a smartphone-based data collection scheme was introduced and successfully tested. In particular, inertial measurement unit (IMU) data collected by accelerometer and gyroscope sensors were used to train supervised machine learning classifiers of different types. The trained models were then used to recognize new examples of construction equipment activities in different levels of detail (LoD).

The primary contribution of the research presented in Chapter 4 was designing and testing a host of high-accuracy supervised machine learning classifiers for recognizing construction equipment activities in different LoD.

In Chapter 5, a wearable data collection framework was designed and implemented for construction workers' activity recognition using built-in sensors of smartphones and machine learning classifiers of different types. The work presented in this Chapter departs from the existing body of knowledge in fields such as computer science, sports, or elderly care considering the fact that task classification in these domains is often performed on certain routine activities such as walking, running, and sitting, whereas in construction operations, recognizing worker activities is not as trivial due to the highly interactive environment of construction jobsites and the freedom of movement each worker has while performing different tasks.

The primary contribution of the research presented in Chapter 5 was designing and testing a detailed framework that used different supervised machine learning classifiers for recognizing complex activities of construction workers in different activity categories and granularities.

In Chapter 6, a complex construction operation was designed that consisted of different processes performed by multiple workers in a large-scale workplace setting. Data were collected from construction workers using smartphones affixed to the upper arms of their working hands. The results of the activity recognition and duration extraction were used inside a data-driven simulation model. The output of this model was compared to the output of a static simulation model, as well as to the observed values in the real world, considering five quantifiable measures. Results indicated that not only does the output of the two simulation models differ, but

also the output of the data-driven simulation model is in closer agreement to the actual values of the measures observed in the real world experiment.

The primary contribution of the research presented in Chapter 6 was extracting activity durations and successfully using them inside a data-driven simulation model to prove the superiority of the developed data-driven simulation modeling framework.

APPENDIX A: COPYRIGHT PERMISSION

As stated in Chapter 1, most of the materials of Chapter 2 have been previously published by the author in the American Society of Civil Engineers (ASCE) Journal of Construction Engineering and Management as a technical paper. The paper is reused in this Dissertation with the following permission from ASCE.

3/9/2015 Rightslink® by Copyright Clearance Center

Copyright Clearance Center RightsLink® Home Create Account Help Live Chat

ASCE AMERICAN SOCIETY OF CIVIL ENGINEERS

Title: Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining

Author: Reza Akhavian, Amir H. Behzadan

Publication: Journal of Construction Engineering and Management

Publisher: American Society of Civil Engineers

Date: 11/01/2013

Copyright © 2013, ASCE. All rights reserved.

Permissions Request

As an ASCE author, you are permitted to reuse you own content for another ASCE or non-ASCE publication.

Please add the full credit line "With permission from ASCE" to your source citation. Please print this page for your records.

Type of use: Dissertation/Thesis

Portion: full article

Format: print and electronic

Use of this content will make up more than 25% of the new work: no

Author of this ASCE work or ASCE will publish the new work: yes

BACK CLOSE WINDOW

Copyright © 2015 Copyright Clearance Center, Inc. All Rights Reserved. [Privacy statement](#) [Terms and Conditions](#). Comments? We would like to hear from you. E-mail us at customerscare@copyright.com

LOGIN

If you're a copyright.com user, you can login to Rightslink using your copyright.com credentials. Already a RightsLink user or want to learn more?

Figure A.1: ASCE permission to reuse the published paper in Chapter 2 of this Dissertation.

APPENDIX B: AUTHOR'S BIOGRAPHY

Reza Akhavian graduated from the University of Tehran with a Bachelor of Science (B.S.) degree in Civil Engineering in 2010. Subsequently he moved to Orlando, Florida, USA to pursue a Master of Science in Civil Engineering (M.S.C.E.) with focus on Construction Engineering and Management at the University of Central Florida where he graduated in 2012. He continued his education at the University of Central Florida and earned a Doctor of Philosophy (Ph.D.) degree in Civil Engineering with focus on Construction Engineering and Management in 2015.

REFERENCES

- [1] J. Martinez, P. Ioannou, General-purpose systems for effective construction simulation, *J. Constr. Eng. Manage.* 125 (4) (1999) 265-276.
- [2] D. Hajjar, S. AbouRizk, Symphony: An Environment for Building Special Purpose Construction Simulation Tools, in: *Proceeding of Proceedings of the 1999 Winter Simulation Conference, ACM, 1999*, pp. 998-1006.
- [3] A.A. Oloufa, Modeling Operational Activities in Object-Oriented Simulation, *J. Comput. Civil Eng.* 7 (1) (1993) 94-106.
- [4] V.R. Kamat, J.C. Martinez, Scene graph and frame update algorithms for smooth and scalable 3D visualization of simulated construction operations, *Journal of Computer-aided civil and infrastructure engineering* 17 (4) (2002) 228-245.
- [5] R. Akhavian, A.H. Behzadan, Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining, *J. Constr. Eng. Manage.* 139 (11) (2013) 04013021.
- [6] M. Lu, Simplified discrete-event simulation approach for construction simulation, *J. Constr. Eng. Manage.* 129 (5) (2003) 537-546.
- [7] S. AbouRizk, J. Shi, Automated construction-simulation optimization, *J. Constr. Eng. Manage.* 120 (2) (1994) 374-385.
- [8] J. Banks, *Handbook of simulation : principles, methodology, advances, applications, and practice*, Wiley, New York, 1998.
- [9] P. Daneshgari, H. Moore, *Electrical Construction and Maintenance, The secret to short-interval scheduling*, Chicago, IL, 2009.
- [10] C.H. Caldas, D.T. Grau, C.T. Haas, Using global positioning system to improve materials-locating processes on industrial projects, *J. Constr. Eng. Manage.* 132 (7) (2006) 741-749.

- [11] W.S. Jang, M.J. Skibniewski, Wireless sensor technologies for automated tracking and monitoring of construction materials utilizing Zigbee networks, in: Proceeding of ASCE Construction Research Congress, Grand Bahamas Island, 2007.
- [12] S. Han, S. Lee, F. Peña-Mora, Application of dimension reduction techniques for motion recognition: construction worker behavior monitoring, in: Proceeding of 2011 ASCE International Workshop on Computing in Civil Engineering, 2011, pp. 19-22.
- [13] A.H. Behzadan, Z. Aziz, C.J. Anumba, V.R. Kamat, Ubiquitous location tracking for context-specific information delivery on construction sites, *Autom. Constr.* 17 (6) (2008) 737-748.
- [14] R. Akhavian, A.H. Behzadan, Dynamic Data Driven Simulation of Ongoing Construction Processes, in: Proceeding of 3rd International and 9th Construction Specialty Conference, Ottawa, Canada, 2011, pp. CN-154 (151-110).
- [15] R. Navon, Automated project performance control of construction projects, *Autom. Constr.* 14 (4) (2005) 467-476.
- [16] I. Brilakis, M.W. Park, G. Jog, Automated vision tracking of project related entities, *Adv. Eng. Inf.* 25 (4) (2011) 713-724.
- [17] D.J.C. MacKay, Information-based objective functions for active data selection, *Neural computation* 4 (4) (1992) 590-604.
- [18] Q. Ling, How Many Low-Precision Sensors are Enough for Reliable Detection?, *Aerospace and Electronic Systems*, *IEEE Transactions on* 47 (4) (2011) 3001-3006.
- [19] A. Véjar, P. Charpentier, Generation of an Adaptive Simulation Driven By Product Trajectories, *Journal of Intelligent Manufacturing* (2011) 1-13.
- [20] Y.J. Son, R.A. Wysk, Automatic Simulation Model Generation for Simulation-Based, Real-Time Shop Floor Control, *Comput. Ind.* 45 (3) (2001) 291-308.
- [21] Y. Yuan, C.A. Dogan, G.L. Viegelaahn, A Flexible Simulation Model Generator, *Comput. Ind. Eng.* 24 (2) (1993) 165-175.

- [22] W.J. Davis, On-Line Simulation: Need and Evolving Research Requirements, in: Handbook of Simulation, John Wiley & Sons, Inc., 2007, pp. 465-516.
- [23] R. Akhavian, A.H. Behzadan, An Integrated Data Collection and Analysis Framework for Remote Monitoring and Planning of Construction Operations, *Adv. Eng. Inf.* (2012).
- [24] S.N. Razavi, C.T. Haas, Reliability-based hybrid data fusion method for adaptive location estimation in construction, *J. Comput. Civil Eng.* 26 (1) (2011) 1-10.
- [25] B. Akinci, F. Boukamp, C. Gordon, D. Huber, C. Lyons, K. Park, A formalism for utilization of sensor systems and integrated project models for active construction quality control, *Autom. Constr.* 15 (2) (2006) 124-138.
- [26] D.T. Grau, C.H. Caldas, Methodology for automating the identification and localization of construction components on industrial projects, *J. Comput. Civil Eng.* 23 (1) (2009) 3-13.
- [27] A. Pradhan, B. Akinci, A taxonomy of reasoning mechanisms and data synchronization framework for road excavation productivity monitoring, *Adv. Eng. Inf.* 26 (2012) 563-573.
- [28] M.W. Park, E. Palinginis, I. Brilakis, Detection of Construction Workers in Video Frames for Automatic Initialization of Vision Trackers, in: *Proceeding of Construction Research Congress 2012@sConstruction Challenges in a Flat World*, ASCE, 2012, pp. 940-949.
- [29] J. Gong, C.H. Caldas, Computer vision-based video interpretation model for automated productivity analysis of construction operations, *J. Comput. Civil Eng.* 24 (3) (2009) 252-263.
- [30] E. Rezazadeh Azar, B. McCabe, Vision-Based Recognition of Dirt Loading Cycles in Construction Sites, in: *Proceeding of ASCE*, 2012, pp. 1042-1051.
- [31] J. Yang, O. Arif, P. Vela, J. Teizer, Z. Shi, Tracking multiple workers on construction sites using video cameras, *Adv. Eng. Inf.* 24 (4) (2010) 428-434.
- [32] L. Soibelman, H. Kim, Data preparation process for construction knowledge generation through knowledge discovery in databases, *J. Comput. Civil Eng.* 16 (2002) 39.

- [33] P. Chen, R.B. Buchheit, J.H. Garrett Jr, S. McNeil, Web-vacuum: Web-based environment for automated assessment of civil infrastructure data, *J. Comput. Civil Eng.* 19 (2005) 137.
- [34] J. Martinez, P.G. Ioannou, General purpose simulation with stroboscope, in: *Proceeding of the 1994 Winter Simulation Conference (WSC)*, Association for Computing Machinery (ACM), Lake Buena Vista, FL, 1994, pp. 1159-1166.
- [35] A.R. Andoh, S. Xing, C. Hubo, A Framework of RFID and GPS for Tracking Construction Site Dynamics, in: *Proceeding of Construction Research Congress 2012@ sConstruction Challenges in a Flat World*, ASCE, 2012, pp. 818-827.
- [36] W.J. Lee, J.H. Song, S.W. Kwon, S. Chin, C. Choi, Y.S. Kim, A gate sensor for construction logistics, in: *Proceeding of Proceedings of the 25th International Symposium on Automation and Robotics in Construction*, 2008, pp. 100-105.
- [37] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, *Advances in knowledge discovery and data mining*, 1996.
- [38] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification and Scene Analysis* 2nd ed, Wiley, 2000.
- [39] C.M. Bishop, *Pattern recognition and machine learning*, springer New York, 2006.
- [40] P. Berkhin, A survey of clustering data mining techniques, *Grouping multidimensional data* (2006) 25-71.
- [41] http://members.tripod.com/asim_saeed/paper.htm, (accessed Jul. 23, 2013).
- [42] B. Mirkin, *Clustering for data mining: a data recovery approach*, Chapman and Hall/CRC, 2005.
- [43] S.C. Mathewson, The Application of Program Generator Software and Its Extensions to Discrete Event Simulation Modeling, *IIE Trans.* 16 (1) (1984) 3-18.

- [44] R. Beheshti, G. Sukthankar, Extracting agent-based models of human transportation patterns, in: Proceeding of Social Informatics (SocialInformatics), 2012 International Conference on, IEEE, 2012, pp. 157-164.
- [45] R. Beheshti, G. Sukthankar, A hybrid modeling approach for parking and traffic prediction in urban simulations, AI & SOCIETY (2014) 1-12.
- [46] D.R. Ford, B.J. Schroer, An expert manufacturing simulation system, Simulation 48 (5) (1987) 193-200.
- [47] G.E. Heidorn, English as a very high level language for simulation programming, in: Proceeding of ACM SIGPLAN Notices, ACM, 1974, pp. 91-100.
- [48] L. Song, C. Cooper, S. Lee, Real-time simulation for look-ahead scheduling of heavy construction projects, in: Proceeding of Construction Research Congress, 2009, pp. 1318-1327.
- [49] T.H. Chung, Y. Mohamed, S.M. AbouRizk, Bayesian Updating Application into Simulation in the North Edmonton Sanitary Trunk Tunnel Project, J. Constr. Eng. Manage. 132 (8) (2006) 882-894.
- [50] J. Sall, A. Lehman, M. Stephens, L. Creighton, JMP start statistics: a guide to statistics and data analysis using JMP, SAS Institute, 2012.
- [51] J.C. Martinez, Stroboscope: State and Resource Based Simulation of Construction Processes, Ph.D Dissertation, University of Michigan, Ann Arbour, MI, 1996.
- [52] R. Akhavian, A.H. Behzadan, Evaluation of queuing systems for knowledge-based simulation of construction processes, Autom. Constr. 47 (2014) 37-49.
- [53] R.B. Cooper, Introduction to queueing theory, North Holland New York, 1981.
- [54] J. Medhi, Stochastic models in queueing theory, Access Online via Elsevier, 2002.
- [55] S. AbouRizk, Role of Simulation in Construction Engineering and Management, J. Constr. Eng. Manage. 136 (10) (2010) 1140-1153.

- [56] J.C. Martinez, Methodology for Conducting Discrete-Event Simulation Studies in Construction Engineering and Management, *J. Constr. Eng. Manage.* 136 (1) (2009) 3-16.
- [57] M. Arashpour, R. Wakefield, N. Blismas, E. Lee, Analysis of disruptions caused by construction field rework on productivity in residential projects, *J. Constr. Eng. Manage.* (2013).
- [58] I. Brodetskaia, R. Sacks, A. Shapira, Stabilizing production flow of interior and finishing works with reentrant flow in building construction, *J. Constr. Eng. Manage.* 139 (6) (2012) 665-674.
- [59] S. Nunnally, *Managing Construction Equipment*, Prentice Hall, New Jersey, 2nd Ed. ed., Prentice-Hall, Upper Saddle River, N.J, 2000.
- [60] D.W. Halpin, L.S. Riggs, *Planning and analysis of construction operations*, Wiley, New York, 1992.
- [61] F. Farid, T.L. Koning, Simulation verifies queuing program for selecting loader-truck fleets, *J. Constr. Eng. Manage.* 120 (2) (1994) 386-404.
- [62] A.M. Law, W.D. Kelton, *Simulation modeling and analysis*, McGraw Hill Boston, MA, 2000.
- [63] P.G. Ioannou, J.C. Martinez, Simulation of Complex Construction Processes, in: *Proceeding of Proceedings of the 28th Conference on Winter Simulation*, Institute of Electrical and Electronics Engineers, Inc, 1996, pp. 1321-1328.
- [64] A.H. Behzadan, V.R. Kamat, Enabling smooth and scalable dynamic 3D visualization of discrete-event construction simulations in outdoor augmented reality, in: *Proceeding of 2007 Winter Simulation Conference (WSC)*, Washington, DC, 2007, pp. 2168-2176.
- [65] S. AbouRizk, D. Halpin, Y. Mohamed, U. Hermann, Research in modeling and simulation for improving construction engineering operations, *J. Constr. Eng. Manage.* 137 (10) (2011) 843-852.
- [66] S. Lee, A. Behzadan, A. Kandil, Y. Mohamed, *Grand Challenges in Simulation for the Architecture, Engineering, Construction and Facility Management Industry*, (2013).

- [67] R. Akhavian, A.H. Behzadan, An Integrated Data Collection and Analysis Framework for Remote Monitoring and Planning of Construction Operations, *Adv. Eng. Inf.* 26 (4) (2012) 749–761.
- [68] R. Akhavian, A.H. Behzadan, Remote Monitoring of Dynamic Construction Processes Using Automated Equipment Tracking, in: *Proceeding of Construction Research Congress 2012*, ASCE, 2012, pp. 1360-1369.
- [69] A. Hammad, C. Zhang, Towards Real-Time Simulation of Construction Activities Considering Spatio-Temporal Resolution Requirements for Improving Safety and Productivity, in: *Proceeding of Proceedings of the 2011 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Inc., 2011, pp. 3533-3544.
- [70] L. Song, N.N. Eldin, Adaptive Real-Time Tracking and Simulation of Heavy Construction Operations for Look-Ahead Scheduling, *Autom. Constr.* 27 (2012) 32-39.
- [71] P.G. Ioannou, Construction of a dam embankment with nonstationary queues, in: *Proceeding of Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future-Volume 2*, ACM, 1999, pp. 921-928.
- [72] L. Darren Graham, S.D. Smith, P. Dunlop, Lognormal distribution provides an optimum representation of the concrete delivery and placement process, *J. Constr. Eng. Manage.* 131 (2) (2005) 230-238.
- [73] M. Oberguggenberger, Queueing models with fuzzy data in construction management, in: *Analyzing Uncertainty in Civil Engineering*, Springer, 2005, pp. 197-210.
- [74] J. Banks, *Discrete Event System Simulation*, Pearson Education, New Jersey, 2005.
- [75] J.F.C. Kingman, *Poisson processes*, Oxford university press, USA, 1992.
- [76] F.R. Jacobs, R.B. Chase, N.J. Aquilano, *Operations and supply management*, McGraw-Hill New York, NY, 2009.

- [77] D.G. Kendall, Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain, *The Annals of Mathematical Statistics* (1953) 338-354.
- [78] J.H. Willenbrock, Estimating costs of earthwork via simulation, *Journal of the Construction Engrg. Division* 98 (1) (1972) 49-60.
- [79] S.M. AbouRizk, D.W. Halpin, J.R. Wilson, Fitting beta distributions based on sample data, *J. Constr. Eng. Manage.* 120 (2) (1994) 288-305.
- [80] C. Maio, C. Schexnayder, K. Knutson, S. Weber, Probability distribution functions for construction simulation, *J. Constr. Eng. Manage.* 126 (4) (2000) 285-292.
- [81] R.G. Miller, Priority queues, *The Annals of Mathematical Statistics* 31 (1) (1960) 86-103.
- [82] L. Takács, Priority Queues, *Oper. Res.* 12 (1) (1964) 63-74.
- [83] J.H. Steffen, J. Hotchkiss, Experimental test of airplane boarding methods, *Journal of Air Transport Management* 18 (1) (2012) 64-67.
- [84] M.H. DeGroot, M.J. Schervish, *Probability and statistics*, 3rd ed., Addison-Wesley, Boston, MA, 2002.
- [85] R. Akhavian, A.H. Behzadan, Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers, *Adv. Eng. Inf.* (2015).
- [86] E.N.R. (ENR), *Eng. News-Rec*, Construction tops trillion dollar annual pace, 2004.
- [87] G.B. Dadi, P.M. Goodrum, T.R. Taylor, C.M. Carswell, Cognitive Workload Demands Using 2D and 3D Spatial Engineering Information Formats, *J. Constr. Eng. Manage.* (2014).
- [88] T. Gao, S. Ergan, B. Akinci, J.H. Garrett, Proactive Productivity Management at Job Sites: Understanding Characteristics of Assumptions Made for Construction Processes during Planning Based on Case Studies and Interviews, *J. Constr. Eng. Manage.* (2013).

- [89] H. Nasir, H. Ahmed, C. Haas, P.M. Goodrum, An analysis of construction productivity differences between Canada and the United States, *Construction Management and Economics* (ahead-of-print) (2013) 1-13.
- [90] J. Gong, C.H. Caldas, Computer vision-based video interpretation model for automated productivity analysis of construction operations, *J. Comput. Civil Eng.* 24 (3) (2010) 252-263.
- [91] G. Locatelli, M. Mancini, G. Gastaldo, F. Mazza, Improving Projects Performance With Lean Construction: State Of The Art, Applicability And Impacts, *Organization, Technology & Management in Construction: An International Journal* 5 (Special) (2013) 775-783.
- [92] G. Ballard, Improving work flow reliability, in: *Proceeding of Proceedings of the 7 th Annual Conference of the International Group for Lean Construction*, 1999, pp. 275-286.
- [93] B. Mccullouch, Automating field data collection in construction organizations, in: *Proceeding of the 1997 ASCE Construction Congress Minneapolis, Minnesota*, 1997, pp. 957-963.
- [94] S.M. AbouRizk, A. Sawhney, Subjective and interactive duration estimation, *Can. J. Civ. Eng.* 20 (3) (1993) 457-470.
- [95] D.Y. Chang, M.S. Hoque, A Knowledge-based Simulation System for Construction Process Planning, in: *Proceeding of Proceedings of the 6th International Symposium on Automation and Robotics in Construction*, 1989, pp. 348-355.
- [96] R. Akhavian, A.H. Behzadan, Automated knowledge discovery and data-driven simulation model generation of construction operations, in: *Proceeding of Simulation Conference (WSC)*, 2013 Winter, IEEE, 2013, pp. 3030-3041.
- [97] Q. Li, J.A. Stankovic, M.A. Hanson, A.T. Barth, J. Lach, G. Zhou, Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information, in: *Proceeding of Wearable and Implantable Body Sensor Networks*, 2009. BSN 2009. Sixth International Workshop on, IEEE, 2009, pp. 138-143.
- [98] K. Van Laerhoven, O. Cakmakci, What shall we teach our pants?, in: *Proceeding of Wearable Computers*, The Fourth International Symposium on, IEEE, 2000, pp. 77-83.

- [99] G. Kochersberger, E. McConnell, M.N. Kuchibhatla, C. Pieper, The reliability, validity, and stability of a measure of physical activity in the elderly, *Archives of physical medicine and rehabilitation* 77 (8) (1996) 793-795.
- [100] E. Alt, Medical interventional device with accelerometer for providing cardiac therapeutic functions, in, Google Patents, 1995.
- [101] K. Altun, B. Barshan, Human activity recognition using inertial/magnetic sensor units, in: *Human Behavior Understanding*, Springer, 2010, pp. 38-51.
- [102] P. Casale, O. Pujol, P. Radeva, Human activity recognition from accelerometer data using a wearable device, in: *Pattern Recognit Image Anal.*, Springer, 2011, pp. 289-296.
- [103] L. Bao, S.S. Intille, Activity recognition from user-annotated acceleration data, in: *Pervasive computing*, Springer, 2004, pp. 1-17.
- [104] N. Ravi, N. Dandekar, P. Mysore, M.L. Littman, Activity recognition from accelerometer data, in: *Proceeding of AAAI*, 2005, pp. 1541-1546.
- [105] S. Namal, A. Senanayake, V. Chong, J. Chong, G. Sirisinghe, Analysis of soccer actions using wireless accelerometers, in: *Proceeding of Industrial Informatics, 2006 IEEE International Conference on*, IEEE, 2006, pp. 664-669.
- [106] J.R. Kwapisz, G.M. Weiss, S.A. Moore, Activity recognition using cell phone accelerometers, *ACM SigKDD Explorations Newsletter* 12 (2) (2011) 74-82.
- [107] K. Motoi, S. Tanaka, M. Nogawa, K. Yamakoshi, Evaluation of a new sensor system for ambulatory monitoring of human posture and walking speed using accelerometers and gyroscope, in: *Proceeding of SICE 2003 Annual Conference*, IEEE, 2003, pp. 1232-1235.
- [108] D.A. Johnson, M.M. Trivedi, Driving style recognition using a smartphone as a sensor platform, in: *Proceeding of Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, IEEE, 2011, pp. 1609-1615.
- [109] P. Lukowicz, J.A. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, T. Starner, Recognizing workshop activity using body worn microphones and accelerometers, in: *Pervasive Computing*, Springer, 2004, pp. 18-32.

- [110] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceeding of Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, IEEE, 2005, pp. 886-893.
- [111] B. Yao, S.-C. Zhu, Learning deformable action templates from cluttered videos, in: Proceeding of Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 1507-1514.
- [112] B. Yao, A. Khosla, L. Fei-Fei, Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses, a) A 1 (D2) (2011) D3.
- [113] J. Zou, H. Kim, Using hue, saturation, and value color space for hydraulic excavator idle time analysis, J. Comput. Civil Eng. 21 (4) (2007) 238-246.
- [114] J. Gong, C.H. Caldas, C. Gordon, Learning and classifying actions of construction workers and equipment using Bag-of-Video-Feature-Words and Bayesian network models, Adv. Eng. Inf. 25 (4) (2011) 771-782.
- [115] M. Golparvar-Fard, A. Heydarian, J.C. Niebles, Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers, Adv. Eng. Inf. 27 (4) (2013) 652-663.
- [116] A.A. Oloufa, M. Ikeda, H. Oda, Situational awareness of construction equipment using GPS, wireless and web technologies, Autom. Constr. 12 (6) (2003) 737-748.
- [117] J. Gong, C.H. Caldas, Data processing for real-time construction site spatial modeling, Autom. Constr. 17 (5) (2008) 526-535.
- [118] E. Ergen, B. Akinci, B. East, J. Kirby, Tracking components and maintenance history within a facility utilizing radio frequency identification technology, J. Comput. Civil Eng. 21 (1) (2007) 11-20.
- [119] A. Montaser, O. Moselhi, RFID+ for tracking earthmoving operations, in: Proceeding of Construction Research Congress, 2012, pp. 1011-1020.

- [120] T. Cheng, M. Venugopal, J. Teizer, P. Vela, Performance evaluation of ultra wideband technology for construction resource location tracking in harsh environments, *Autom. Constr.* 20 (8) (2011) 1173-1184.
- [121] R. Akhavian, A.H. Behzadan, Client-Server Interaction Knowledge Discovery for Operations-level Construction Simulation Using Process Data, in: *Proceeding of Construction Research Congress, ASCE, 2014*, pp. 41-50.
- [122] M. Gul, F. Necati Catbas, Statistical pattern recognition for Structural Health Monitoring using time series modeling: Theory and experimental verifications, *Mech. Syst. Sig. Process.* 23 (7) (2009) 2192-2204.
- [123] J.P. Lynch, K.J. Loh, A summary review of wireless sensors and sensor networks for structural health monitoring, *Shock and Vibration Digest* 38 (2) (2006) 91-130.
- [124] G.W. Roberts, X. Meng, A.H. Dodson, Integrating a global positioning system and accelerometers to monitor the deflection of bridges, *J. Surv. Eng.* 130 (2) (2004) 65-72.
- [125] M. Malekzadeh, M. Gul, F.N. Catbas, Use of FBG Sensors to Detect Damage from Large Amount of Dynamic Measurements, in: *Topics on the Dynamics of Civil Structures, Volume 1*, Springer, 2012, pp. 273-281.
- [126] T. Cheng, G. Migliaccio, J. Teizer, U. Gatti, Data fusion of real-time location sensing (RTLs) and physiological status monitoring (PSM) for ergonomics analysis of construction workers, *ASCE Journal of Computing in Civil Engineering* (2012).
- [127] L. Joshua, K. Varghese, Accelerometer-based activity recognition in construction, *J. Comput. Civil Eng.* 25 (5) (2011) 370-379.
- [128] C.R. Ahn, S. Lee, F. Peña-Mora, The Application of Low-Cost Accelerometers for Measuring the Operational Efficiency of a Construction Equipment Fleet, *J. Comput. Civil Eng.* (2013).
- [129] D.W. Halpin, CYCLONE - A Method for Modeling Job Site Processes, *ASCE Journal of Construction Division* 103 (3) (1977) 489-499.

- [130] P. Ioannou, UM-CYCLONE User's Guide, Dept. of Civil Engineering, The University of Michigan, Ann Arbor, Mich (1989).
- [131] D.W. Halpin, Microcyclone User's Manual, Division of Construction Engineering and Management, Purdue University, West Lafayette, Indiana (1990).
- [132] C.C. Douglas, C.E. Shannon, Y. Efendiev, R. Ewing, V. Ginting, R. Lazarov, M.J. Cole, G. Jones, C.R. Johnson, J. Simpson, A Note on Data-Driven Contaminant Simulation, in: Proceeding of Computational Science - ICCS 2004, Krakow, Poland, 2004, pp. 701-708.
- [133] F. Darema, Dynamic Data Driven Applications Systems: New Capabilities for Application Simulations and Measurements, in: Proceeding of Computational Science -- ICCS 2005, Atlanta, USA, 2005, pp. 610-615.
- [134] C.C. Douglas, Y. Efendiev, A Dynamic Data-Driven Application Simulation Framework for Contaminant Transport Problems, *Journal of Computers & Mathematics with Applications* 51 (11) (2006) 1633-1646.
- [135] S.-Y. Lin, K.-M. Chao, C.-C. Lo, Service-oriented dynamic data driven application systems to traffic signal control, in: Proceeding of Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on, IEEE, 2010, pp. 3463-3470.
- [136] Y. Huang, A. Verbraeck, A dynamic data-driven approach for rail transport system simulation, in: Proceeding of Winter Simulation Conference, Winter Simulation Conference, 2009, pp. 2553-2562.
- [137] J. Tannock, B. Cao, R. Farr, M. Byrne, Data-driven simulation of the supply-chain—insights from the aerospace sector, *Int. J. Prod. Econ.* 110 (1) (2007) 70-84.
- [138] F. Vahdatikhaki, A. Hammad, Framework for near real-time simulation of earthmoving projects using location tracking technologies, *Autom. Constr.* 42 (2014) 50-67.
- [139] M. Lu, F. Dai, W. Chen, Real-time decision support for planning concrete plant operations enabled by integrating vehicle tracking technology, simulation, and optimization algorithms, *Can. J. Civ. Eng.* 34 (8) (2007) 912-922.

- [140] V.R. Kamat, Visualizing simulated construction operations in 3D, *J. Comput. Civil Eng.* 15 (4) (2001) 329-327.
- [141] V.R. Kamat, VITASCOPE: Extensible and scalable 3D visualization of simulated construction operations, Ph.D Dissertation, University Libraries, Virginia Polytechnic Institute and State University, 2003.
- [142] R. Akhavian, A.H. Behzadan, Dynamic Simulation of Construction Activities Using Real Time Field Data Collection, in: *Proceeding of 18th Workshop of Intelligent Computing in Engineering and Architecture (EG-ICE)*, Enschede, Netherlands, 2011, pp. 1-10.
- [143] R. Want, An introduction to RFID technology, *Pervasive Computing, IEEE* 5 (1) (2006) 25-33.
- [144] G. Lyons, K. Culhane, D. Hilton, P. Grace, D. Lyons, A description of an accelerometer-based mobility monitoring technique, *Med. Eng. Phys.* 27 (6) (2005) 497-504.
- [145] D. Figo, P.C. Diniz, D.R. Ferreira, J.M. Cardoso, Preprocessing techniques for context recognition from accelerometer data, *Personal and Ubiquitous Computing* 14 (7) (2010) 645-662.
- [146] M. Khan, S.I. Ahamed, M. Rahman, R.O. Smith, A feature extraction method for realtime human activity recognition on cell phones, in: *Proceeding of Proceedings of 3rd International Symposium on Quality of Life Technology (isQoLT 2011)*. Toronto, Canada, 2011, pp.
- [147] R.W. DeVaul, S. Dunn, Real-time motion classification for wearable computing applications, 2001, project paper, <http://www.media.mit.edu/wearables/mithril/realtime.pdf> (2001).
- [148] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: *Proceeding of ICML, 2003*, pp. 856-863.
- [149] S. Pirttikangas, K. Fujinami, T. Nakajima, Feature selection and activity recognition from wearable sensors, in: *Ubiquitous Computing Systems*, Springer, 2006, pp. 516-527.
- [150] M.A. Hall, Correlation-based feature selection for machine learning, Ph.D Dissertation, The University of Waikato, 1999.

- [151] D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-wesley Reading Menlo Park, 1989.
- [152] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, ACM Sigkdd Explorations Newsletter 6 (1) (2004) 1-6.
- [153] P. Xanthopoulos, T. Razzaghi, A weighted support vector machine method for control chart pattern recognition, Comput. Ind. Eng. 70 (2014) 134-149.
- [154] T.G. Dietterich, Ensemble methods in machine learning, in: Multiple classifier systems, Springer, 2000, pp. 1-15.
- [155] R. Vilalta, Y. Drissi, A perspective view and survey of meta-learning, Artificial Intelligence Review 18 (2) (2002) 77-95.
- [156] T. Cheng, J. Teizer, G.C. Migliaccio, U.C. Gatti, Automated task-level activity analysis through fusion of real time location sensors and worker's thoracic posture data, Autom. Constr. 29 (2013) 24-39.
- [157] J. Gong, C.H. Caldas, An intelligent video computing method for automated productivity analysis of cyclic construction operations, in: Proceeding of Proc., the 2009 ASCE International Workshop on Computing in Civil Engineering, 2009, pp. 64-73.
- [158] C. Kim, T. Park, H. Lim, H. Kim, On-site construction management using mobile computing technology, Autom. Constr. 35 (2013) 415-423.
- [159] C. De Dominicis, A. Depari, A. Flammini, S. Rinaldi, E. Sisinni, Smartphone based localization solution for construction site management, in: Proceeding of Sensors Applications Symposium (SAS), 2013 IEEE, IEEE, 2013, pp. 43-48.
- [160] L. Chen, J. Hoey, C.D. Nugent, D.J. Cook, Z. Yu, Sensor-based activity recognition, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 42 (6) (2012) 790-808.

- [161] V. Escorcia, M. Dávila, M. Golparvar-Fard, J. Niebles, Automated vision-based recognition of construction worker actions for building interior construction operations using RGBD cameras, in: Proceeding of Construction Research Congress, 2012, pp. 879-888.
- [162] J.Y. Kim, C.H. Caldas, VISION-BASED ACTION RECOGNITION IN THE INTERNAL CONSTRUCTION SITE USING INTERACTIONS BETWEEN WORKER ACTIONS AND CONSTRUCTION OBJECTS, in: Proceeding of 30th International Symposium on Automation and Robotics in Construction, 2013, pp. 1-8.
- [163] A. Peddi, L. Huan, Y. Bai, S. Kim, Development of human pose analyzing algorithms for the determination of construction productivity in real-time, in: Proceeding of Construction Research Congress Seattle, WA, 2009, pp. 11-20.
- [164] R. Gonsalves, J. Teizer, Human motion analysis using 3D range imaging technology, in: Proceeding of Proceedings of the 26th International Symposium on Automation and Robotics in Construction, Austin, Texas, 2009, pp. 76-85.
- [165] S. Han, S. Lee, A vision-based motion capture and recognition framework for behavior-based safety management, *Autom. Constr.* 35 (2013) 131-141.
- [166] T. Cheng, G.C. Migliaccio, J. Teizer, U.C. Gatti, Data fusion of real-time location sensing and physiological status monitoring for ergonomics analysis of construction workers, *J. Comput. Civil Eng.* 27 (3) (2012) 320-335.
- [167] U. Maurer, A. Smailagic, D.P. Siewiorek, M. Deisher, Activity recognition and monitoring using multiple sensors on different body positions, in: Proceeding of Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on, IEEE, 2006, pp. 4 pp.-116.
- [168] T. Brezmes, J.-L. Gorricho, J. Cotrina, Activity recognition from accelerometer data on a mobile phone, in: Distributed computing, artificial intelligence, bioinformatics, soft computing, and ambient assisted living, Springer, 2009, pp. 796-799.
- [169] L. Sun, D. Zhang, B. Li, B. Guo, S. Li, Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations, in: Ubiquitous intelligence and computing, Springer, 2010, pp. 548-562.

- [170] A. Bayat, M. Pomplun, D.A. Tran, A Study on Human Activity Recognition Using Accelerometer Data from Smartphones, *Procedia Computer Science* 34 (2014) 450-457.
- [171] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M. Srivastava, Using mobile phones to determine transportation modes, *ACM Transactions on Sensor Networks (TOSN)* 6 (2) (2010) 13.
- [172] T.-S. Kim, J.-H. Cho, J.T. Kim, Mobile motion sensor-based human activity recognition and energy expenditure estimation in building environments, in: *Sustainability in Energy and Buildings*, Springer, 2013, pp. 987-993.
- [173] H. Martín, A.M. Bernardos, J. Iglesias, J.R. Casar, Activity logging using lightweight classification techniques in mobile devices, *Personal and ubiquitous computing* 17 (4) (2013) 675-695.
- [174] X. Su, H. Tong, P. Ji, Activity recognition with smartphone sensors, *Tsinghua Science and Technology* 19 (3) (2014) 235-249.
- [175] M. Shoaib, S. Bosch, O.D. Incel, H. Scholten, P.J. Havinga, A Survey of Online Activity Recognition Using Mobile Phones, *Sensors* 15 (1) (2015) 2059-2085.
- [176] L. Joshua, K. Varghese, Selection of Accelerometer Location on Bricklayers Using Decision Trees, *Computer-Aided Civil and Infrastructure Engineering* 28 (5) (2013) 372-388.
- [177] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, K. Aberer, Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach, in: *Proceeding of Wearable Computers (ISWC)*, 2012 16th International Symposium on, Ieee, 2012, pp. 17-24.
- [178] M. Kose, O.D. Incel, C. Ersoy, Online human activity recognition on smart phones, in: *Proceeding of Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data*, 2012, pp. 11-15.
- [179] M. Berchtold, M. Budde, D. Gordon, H.R. Schmidtke, M. Beigl, Actiserv: Activity recognition service for mobile phones, in: *Proceeding of Wearable Computers (ISWC)*, 2010 International Symposium on, IEEE, 2010, pp. 1-8.

- [180] S. Chernbumroong, A.S. Atkins, H. Yu, Activity classification using a single wrist-worn accelerometer, in: Proceeding of Software, Knowledge Information, Industrial Management and Applications (SKIMA), 2011 5th International Conference on, IEEE, 2011, pp. 1-6.
- [181] J. Staudenmayer, D. Pober, S. Crouter, D. Bassett, P. Freedson, An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer, *Journal of Applied Physiology* 107 (4) (2009) 1300-1307.
- [182] S.S. Haykin, S.S. Haykin, S.S. Haykin, S.S. Haykin, *Neural networks and learning machines*, Pearson Education Upper Saddle River, 2009.
- [183] M.H. Hassoun, *Fundamentals of artificial neural networks*, MIT press, 1995.
- [184] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, R. Tibshirani, *The elements of statistical learning*, Springer, 2009.
- [185] OSHA, Safety and Health Administration, Excavation Final Rule, U.S.D.o. Labor, 1990.
- [186] <http://www.cdc.gov/niosh/topics/safety.html>, (accessed March 30, 2015).
- [187] C.I.I. (CII), Construction Industry Institute, *Productivity measurement: An introduction*, T.U.o.T.a. Austin, Austin, Tex., 1990.
- [188] H.-S. Park, S.R. Thomas, R.L. Tucker, Benchmarking of construction productivity, *J. Constr. Eng. Manage.* 131 (7) (2005) 772-778.
- [189] H.R. Thomas, Labor productivity and work sampling: the bottom line, *J. Constr. Eng. Manage.* 117 (3) (1991) 423-444.
- [190] H.R. Thomas, J. Daily, Crew performance measurement via activity sampling, *J. Constr. Eng. Manage.* 109 (3) (1983) 309-320.
- [191] U.C. Gatti, G.C. Migliaccio, S.M. Bogus, S. Schneider, An exploratory study of the relationship between construction workforce physical strain and task level productivity, *Construction Management and Economics* 32 (6) (2014) 548-564.

[192] L.K. Hansen, P. Salamon, Neural network ensembles, *IEEE Trans. Pattern Anal. Mach. Intell.* 12 (10) (1990) 993-1001.

[193] X. Lin, S. Yacoub, J. Burns, S. Simske, Performance analysis of pattern classifier combination by plurality voting, *Pattern Recognit. Lett.* 24 (12) (2003) 1959-1969.