NATURE-INSPIRED SENSOR DATA ANALYTICS METHODS FOR SIMULATION INPUT MODELING

A Thesis

by

PRABHAT SHRESTHA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Amir H. Behzadan
Committee Members,	Changbum Ryan Ahn
	Theodora Chaspari
	Eric Jing Du
Head of Department,	Patrick Suermann

May 2018

Major Subject: Construction Management

ABSTRACT

The general proliferation of technology including smartphones and sensors present both an opportunity and a challenge for the construction industry. On the one hand, it creates an opportunity for improved efficiency via greater data-driven decision-making, but on the other hand, presence of noise and uncertainty in the captured data (due to the dynamic and intermittent nature of construction processes), pose significant hurdles to widespread adoption and utilization. Moreover, there is a dearth of domain-specific research concerning the systematic treatment and elimination of such noise. This can have significant impact in the output. As the chaos theory explains, initial noise (even in small portions) can prove to be detrimental to the overall efficacy of a system due to the volatility induced by propagation of such noise through the system. Most natural systems, however, maintain stability and improve over time. In particular, species have improved with evolution, and complex biological information have been preserved and transferred through DNA coding and utilized effectively across generations. Thus, the hypothesis of this research is that methodologies based on principles of natural phenomena can enable reliability of the collected sensor data. This hypothesis is validated by processing data through genetic algorithms (GA), sequence alignment (SA), and multi-dimensional sequence alignment (MSA), all rooted in nature. Processed data is then used to create key input for simulation models describing the real system. Findings of this work is sought to provide project managers and stakeholders with better insights into the nature of crew

activities and interactions, and help select the most effective combination of resources while reducing the amount and frequency of rework.

DEDICATION

I dedicate this Thesis

to

my mother, Indira Devi Shrestha and my father, Kedar Man Shrestha

they inspire me to strive to be better

&

show me how to everyday

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to Dr. Amir Behzadan who gave me this amazing opportunity and guided me through the thick and thin. He continuously challenged me, inspired me to meet those challenges, and guided me through the challenges. I am a better researcher because of this. I would also like to thank Dr. Changbun Ryan Ahn, Dr. Theodora Chaspari and Dr. Eric Jing Du for serving on my committee and for their invaluable suggestions. Their inputs were instrumental in improving this Thesis.

I would also like to thank the wonderful fellow researchers at CIBER Lab, Nipun Nath, Songjukta Dutta, and Khandakar Rashid for their thoughts, for listening to my half-formed ideas, and for their support and encouragement. I am thankful for my parents who taught me the principles of life and the value of education. Their sacrifices, integrity, and perseverance inspire me every day. Finally, I am grateful to my brother, Pranav Shrestha who has been a wonderful source of support throughout this journey.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a Thesis committee consisting of Professor Amir H. Behzadan [advisor], Professor Changbun Ryan Ahn and Professor Eric Jing Du of the Department of Construction Science, and Professor Theodora Chaspari of the Department of Computer Science at Texas A&M University.

The parallel work on human activity recognition (HAR) depicted sparsely throughout this Thesis, and used to draw comparisons and conclusions was conducted in part by Mr. Nipun Nath, doctoral student in the Department of Construction Science, and a researcher in the Construction Informatics and Built Environment Research (CIBER) Lab. All other work conducted as part of this Thesis was completed by the author independently.

This work has been supported by the U.S. National Science Foundation (NSF) through grants CMMI 1602236 and CMMI 1800957. The author gratefully acknowledges the support from the NSF. Any opinions, findings, conclusions, and recommendations expressed in this Thesis are those of the author and do not necessarily represent those of the NSF.

NOMENCLATURE

ACD	Activity Cycle Diagram
AEC/FM	Architecture, Engineering, Construction/Facility Management
CCM	Cumulative Confusion Matrix
CI	Confidence Interval
DES	Discrete Event Simulation
DNA	Dependency Network Assimilator/ Deoxyribonucleic Acid
FDs	False Detections
FP	False Positive
FN	False Negative
GA	Genetic Alignment
GFP	Global Fitness Parameter
HAR	Human Activity Recognition
HMM	Hidden Markov Model
HPRC	High Performance Research Center
MSA	Multi-Dimensional Sequence Alignment
RNA	Ribonucleic Acid
SA	Sequence Alignment
TP	True Positive
TN	True Negative

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

I.1 Background

The construction industry is one of the major sectors of the U.S. economy with the total spending of the industry estimated to be approximately $1.2 billion in 2017 (U.S. Census Bureau 2017). It also employs about 9 million workers accounting for about 6% of the entire U.S. workforce (CPWR 2016). Despite this enormous footprint in the nation's economy, the construction industry has traditionally been very slow to adapt to and utilize new technology advancements, and incorporate new knowledge areas into its business practices (Becerik-Gerber et al. 2011; World Economic Forum 2016), causing this industry to lag behind in efficiency and productivity growth (U.S. Department of Commerce 2014). Furthermore, recent studies have shown that about 75% of construction projects fail to finish on time and within budget (KPMG 2015). Among other reasons, this could be due to the fact that most construction schedules are subject to uncertainties in durations or activity sequences. Such variability in schedule and resource availability leads to work interruptions, inefficient processes and workflows, and redundancy of effort due to rework and imperfect information (Assaf et al. 1995; Rosenfeld 2014). While each project is different, studies suggest that about 37% of the assumptions made in the initial

planning phases of a construction project turn out to be invalid once the project is launched (Gao et al. 2013).

In the past few decades, simulation modeling has been proposed as a remedy to help identify project uncertainties and their impact on the overall project execution. Simulation models allow project planners to run a large number of possible scenarios, identify the best and worst cases, and design and implement appropriate contingencies for each case ahead of time, all in advance of committing real resources to the project. In order to utilize simulations in the uncertain, dynamic, and transient environment in which the majority of construction projects takes place, it is imperative to use the most reliable input information in order to increase the reliability and applicability of the simulation results. To this end, there is a need for a practical approach to timely collection of data describing the true status of a project, efficiently processing and simulating such data, and meaningfully presenting the results to project stakeholders to support data-driven decision-making. Such an integrated framework would also enable project managers to select the most effective combination of resources (i.e. equipment, labor, and materials) while reducing the amount and frequency of rework.

While simulation has been traditionally utilized for various applications in the project planning and design (Carr 1979; Martinez and Ioannou 1994), pre-construction (Azhar et al. 2008; Portas and AbouRizk 1997), and operation and maintenance (AbouRizk et al. 2011; Marzouk and Moselhi 2003), its implementation in the construction phase has to a large extent remained limited. Previous studies as well as the

author's investigation of the root causes of this issue have revealed that the constantly evolving ground truth in active construction sites is a major impediment to robust and timely data collection, a key precursor of data-driven simulation modeling (Akhavian 2015; Leite et al. 2016; Shrestha and Behzadan 2017).

In light of these challenges, the major theme of this Thesis is the utilization of advances in mobile sensing and data mining to facilitate simulation-based decision-making in construction. In a nutshell, this document will report on a systematic study conducted by the author to utilize built-in smartphone sensors for continuous field data collection, eliminate and/or reduce unwanted noise in collected data using techniques inspired by data-intensive natural systems, recognize human activities through data mining, and use the results to generate simulate models describing field activities with improved accuracy.

This Chapter introduces some of the concepts used to build a framework for data processing that primarily deals with noise.

I.2 Sensors and simulations in construction

Advances in data capturing, processing, and transmission technologies in recent years have proliferated the number and types of sensors available for general use. The processing capabilities of computers have also expanded significantly. Among various types of sensors, wearable (i.e. mobile) sensors are being increasingly used for their ubiquity, affordability, unobtrusiveness, and ease of use (Chen and Khalil 2011).

Prior to the invention and widespread use of sensors and mobile technology at the consumer level, simulation systems had been used as a medium to model the variability and dynamic nature of engineering systems. Within the construction domain, process simulation has been used for project scheduling (Martinez and Ioannou 1994), productivity analysis (Portas and AbouRizk 1997), and mitigating operational conflicts (AbouRizk et al. 2011). By simulating the various possible combinations of events, worst-case scenarios and best-case strategies can be identified. As new data become available, such simulations can be updated and appropriate plans of action adapted to yield better results. More recently, computational frameworks that take advantage of sensor data collection and processing, data mining, and simulation modeling have been proposed as a promising solution to some of the long-standing decision-making problems in construction, such as the inability to integrate execution-phase data into decision-making (Akhavian and Behzadan 2013a; RazaviAlavi and AbouRizk 2016).

I.3 The problem of noise in sensor data

The proliferation of sensors has led to great quantities of collected data with different quality, resolution, and attributes. Surprisingly, this abundance in data quantity has also led to gaps in data utilization, overlooking useful data, or reaching contradicting conclusions depending on how data from different sources are interpreted by the end user based on his/her perception, bias, expectations, skills, or training. In addition, not all collected data is of expected quality and resolution. The relatively high upfront investment

(procurement, installation, and maintenance) cost of sensing technologies often encourages only the adoption and use of low-cost sensors, especially in industries with narrow profit margin such as construction. A major implementation challenge in working with data particularly as related to harsh and dynamic construction environments is the uncertainty inherent to the collected data, which is inevitable in low-cost sensor networks.

To the most extent, generated data is not fully utilized due to issues such as the lack of computationally efficient processing frameworks, high upfront costs, data loss, latency, and reliability issues (Islam et al. 2012), as well as noise and human errors in data collection and mining (Zamalloa and Krishnamachari 2007).

In particular, noise in the collected data reduces the reliability of the conclusions drawn from data and thus, increases the hesitancy to use that information to base decisions. Therefore, with the objective of enabling greater utilization of sensor data in simulation modeling, the research conducted in this Thesis aims at designing a scientific methodology that helps increase the reliability of sensor data through a systematic approach to noise reduction and/or elimination.

I.4 Propagation of noise in sensor readings through the lens of chaos theory

The resulting volatility from imperfect sensor data can be described using the chaos theory, which is the study of complex, nonlinear, dynamic systems (Lorenz 1963). Chaos theory is a branch of mathematics that deals with systems that have the appearance of being deterministic (e.g. a construction schedule) but can experience chaotic events

(e.g. random variations). The theory explains that despite its deterministic nature, a dynamic system that is highly sensitive to initial conditions can behave in a very unpredictable (i.e. chaotic) manner. The presence of this chaos was first observed in the "Lorenz" system which was a set of three ordinary differential equations that described atmospheric convection. Despite their simple form with a determined solution, it was found that the final result varied significantly due to changes in the initial conditions. This variation is the reason why even with the most powerful computers, future patterns in weather systems cannot be predicted beyond a limited time frame. Similarly, other natural phenomena also incorporate elements of chaos. For example, the loose dependence of discrete population models on the initial conditions has been explained by Liz and Ruis-Harrera (2012) as an implementation of chaos theory.

Within the scope of this work, the implication of chaos theory is that if uncertain data from a sensor network is used to build models of a dynamic construction system (even if the actual system appears linear and deterministic), the performance of the model can randomly change with a small variation in initial conditions (i.e. accuracy of sensor readings).

I.5 Natural phenomena that deal with noisy data

In order to achieve the goal of utilizing collected data despite the presence of inherent noise, various algorithms from other domains that deal with noise in data are

examined in this research. In particular, phenomena in nature are of interest due to the unmatched capability of (data-intensive) natural systems to thrive amidst significant noise.

I.5.1 Genetic algorithms

Genetic algorithms (GAs) (Reeves 2003) is a general name given to a family of evolutionary data processing algorithms inspired by the natural selection process observed during biological evolution. Like in the nature, a GA gradually improves the overall population characteristics by invoking operations such as selection (of the best species), crossover (of two or multiple species), and mutation (of parts of a species). GAs have been used in the past in different disciplines such as water contamination characterization (Preis and Ostfeld 2008), evaluating construction plans using data environment analysis (Torabi and Mahlooji 2017), site layout planning for construction projects (RazaviAlavi and AbouRizk 2016), and speech recognition based on random projections (Kataoka et al. 2016).

In general, a GA-based method uses five key operations to reach an optimal solution from a number of possible (but not optimal) solutions (Poli et al. 2008). First, the initial population is taken as the mother generation. After selecting a predetermined portion of this population, the daughter generation is produced through mating among the mother species. The daughter species are evaluated using a predefined fitness function, and this iterative process continues until a desirable stopping condition is met. These principles are adapted and combined with simulations in the context of this research, with findings discussed in detail in Chapter II of this Thesis.

I.5.2 Sequence alignment

Sequence alignment (SA) is a well-established technique in bioinformatics for analyzing deoxyribonucleic acid (DNA), ribonucleic acid (RNA), or protein sequences and identifying regions of similarity. The main goal of SA is to discover relationships between strings of data by deploying a series of heuristic or probabilistic methods to align a new string (e.g. DNA of a new species) with an existing string (DNA of a known species). SA has also been used sporadically in linguistics (Barzilay and Lee 2003), social sciences (Abbott and Tsay 2000), and human resource functions (Blair-Loy 1999). Traditional quantitative measures such as data clustering use a point-by-point approach to analyze sequences (Abbott 1995). This, however, can quickly turn into an exponentially complex problem as each new data point is possibly a point of diversion where a new parallel problem with equal complexity is created. The SA technique tries to remedy these issues by dealing with sequences as a whole. SA measures the degree of similarity between two sequences (a.k.a. "source" and "target" sequences), using three basic operations, namely deletion (where an element is removed from the target sequence), insertion (where an element is added to the target sequence), and substitution (where two elements are switched in the target sequence) (Shoval and Isaacson 2007).

I.5.3 Multi-dimensional sequence alignment (MSA)

Multi-dimensional sequence alignment (MSA) is an expanded form of SA where the sequences are simultaneously compared across several attributes. In general, data streams are evaluated against the ground truth considering their multiple dimensions (e.g.

three in case of a three-dimensional data point), in a multi-dimensional holistic comparison scheme. Potentially, this can increase the depth of insights garnered from a multi-dimensional dataset (Elias 2006), as it opens the door to incorporating more contextual information when making a determination about the fitness of individual data points contained within a much larger dataset. Moving from traditional SA to MSA also enables the transition from activity-level data (post-processing) to sensor-level data (pre-processing), where multiple distinctive data features can serve as dimensions for SA analysis.

I.6 Research objectives and contributions

Despite unparalleled improvements in computing and sensor technologies, the presence of noise in captured data is still preventing the full utilization of sensors in architecture, engineering, and construction (AEC) domains. In general, the current body of knowledge does not support a comprehensive framework that correctly identifies and processes data while dealing with the inherent noise. In order to help mitigate this gap in knowledge, a comprehensive data processing framework that can handle noise in the data collected is required. Thus, the working hypothesis of this research is that nature-inspired techniques can improve current methods of generating discrete event simulation (DES) input models from raw sensor data beyond what is currently achievable by pure computational methods such as human activity recognition (HAR). Such improvement can be described in terms of better quality of simulation input data, closer resemblance of

simulation output to ground-truth information, decreased algorithm processing time, ability to factor in domain-specific parameters and constraints, or a combination of these measures. In particular, two major categories of natural phenomena are investigated in this research, with results documented in this Thesis. These include evolutionary techniques (i.e. genetic algorithms) and SA (both pairwise and multi-dimensional).

This Thesis introduces and validates several algorithms that can be implemented in order to achieve this objective. Natural phenomena, despite being data-intensive, have been successfully dealing with imperfections and noise in data, and producing improved overall populations across multiple generations. The applicability of such nature-inspired methods to refine imperfect sensor data captured by mobile devices (i.e. smartphones) is demonstrated in this Thesis, with the ultimate goal of promoting simulation-based decision-making by reducing the technical expertise and upfront cost of data acquisition using consumer-grade sensors.

While GA techniques have been studied (in other contexts) rather extensively within the AEC domains, the depth and breadth of the body of knowledge around newer methods such as SA and MSA is almost non-existent. In line with this, the materials presented in Chapters II, III and IV of this Thesis specifically seek to create and test new methods that allow for a high-fidelity transformation of raw sensor data into contextual knowledge. Such knowledge can be in part used to describe the status and sequence of activities that take place in a dynamic engineering system, while also providing a basis for

performance benchmarking and identifying areas of waste, mistakes, and inefficiencies within the system.

Overall, this Thesis contributes to the body of knowledge and practice in the construction domain by introducing and validating a host of algorithmic approaches for transforming imperfect raw sensor data into contextual knowledge, incorporating such computer-interpretable knowledge into data-driven simulation models, and generating high-fidelity outputs for better and more reliable execution-phase decision-making.

I.7 Organization of the thesis

This Thesis is divided into five main Chapters. A brief introduction of each Chapter is provided in the following paragraphs.

Introduction. In this Chapter, the problem statement, background information, research motivation, and research objectives are described.

Improving activity recognition using genetic algorithms. In this Chapter, the collection of human time-motion data from a warehouse operation experiment using built-in smartphone sensors (accelerometer, linear accelerometer, and gyroscope) is described. Collected data is first processed through a HAR algorithm to identify transitions between successive activities. Next, results are compared with the ground truth and errors are significantly reduced using a GA-enabled simulation model.

Improving activity dependency data using sequence alignment. Similar to the methodology used in the previous Chapter, this Chapter describes the post-processing of

sensor data using SA. In particular, human time-motion data is first processed through HAR to recognize field activities. Results (which contain errors) are then used as input to SA which in turns uses ground truth data as a template to eliminate and/or reduce inconsistencies in activity sequences.

Refining sensor level data using multi-dimensional sequence alignment (MSA). This Chapter describes the implementation of MSA algorithm applied directly to raw sensor readings. Traditional SA (that is best suited for one-dimensional data stream comparison) fails to work with raw sensor data, as each data point spans over multiple dimensions. In contrast, MSA enables pre-processing of sensor data as it simultaneously processes multiple dimensions of each data point.

Conclusions and future work. This Chapter summarizes the materials and discussions presented in this Thesis, articulates key findings of this research, and provides closing remarks on the contributions of this study to the body of knowledge and practice, as well as potential directions of future work.

CHAPTER II

IMPROVING ACTIVITY RECOGNITION USING GENETIC ALGORITHMS*

II.1 Introduction

II.1.1 Value of simulation to project planning

In most projects, the exact sequence of tasks (or events) cannot be predetermined as only the general precedence logic is known and the execution of tasks is subject to variation and interchanges due to a number of factors. For example, finishing tasks in construction is a large family of activities without co-dependencies. As an example, while flooring and painting cannot be conducted simultaneously at the same location, their order is interchangeable. This inherent variation can have large implication in the overall project execution and performance. According to Park (2006), factors such as overtime, change orders, material management, weather, and human factors cause productivity and schedule variations and create uncertainties in in project performance. In order to mitigate the difficulty of representing the effect of these factors, DES was introduced as an effective solution to deal with the complexity of mathematical modeling and representation of the

* Parts of this chapter have been previously published in "Chaos Theory–Inspired Evolutionary Method to Refine Imperfect Sensor Data for Data-Driven Construction Simulation" by Prabhat Shrestha and Amir H. Behzadan, Journal of Construction Engineering and Management, 3, 144, Copyright [2018] by ASCE, and have been reused with permission from ASCE. This material may be downloaded for personal use only. Any other use requires prior permission of the American Society of Civil Engineers. This material may be found at https://ascelibrary.org/doi/10.1061/(ASCE)CO.1943-7862.0001441

dynamic environments. (Lin and Ying 2002). DES has been used as an effective tool in modeling uncertainties for better project planning and implementation (Martinez and Ioannou 1997). In addition to construction domain-specific platforms, simulation tools have also been widely used in defense, operations research, and logistics (Law et al. 1991). The value of DES is borne out of its ability to produce models that can mimic complex dynamic systems, (Lin and Ying 2002; Skoogh et al. 2012). This is of great use in construction where the start and end times of activities are discrete events with intrinsic uncertainty (AbouRizk et al. 2011; Akhavian and Behzadan 2016; Jang and Skibniewski 2009; Nath 2017).

Despite the great benefits that can be harnessed from it, simulation-based decision-making is often underutilized in practice. Factors such as lack of flexibility (a.k.a. rigidity) of the simulation model, user incompetence, and specificity of the simulation environment (Hajjar and AbouRizk 2002) prevent full use. This is compounded by the need for time and effort in initial conceptualization and formulation of simulation models (Oloufa et al. 1998) and inability of most models to receive and process live construction phase data (Leite et al. 2016). This proliferates the impression that simulation models are difficult to set up while providing limited benefits.

With the advent of faster and more reliable data collection and processing, the integration of field data into simulation models has been investigated actively in recent years. Such studies, however, have been mostly carried out in fields outside architecture, engineering, construction, and facility management (AEC/FM). Akhavian and Behzadan

(2013b) identify some of the efforts in dynamic data-driven application simulation (DDDAS) as used in traffic engineering (Lin et al. 2010), railway engineering simulation (Huang and Verbraeck 2009), and supply change modeling in aerospace engineering (Tannock et al. 2007). Furthermore, the work also describes limited efforts in construction on data-driven simulation. These approaches are to a large extent focused on equipment location data (Song and Eldin 2012).

In this regard, innovative applications of merging live information with dynamic simulations are being explored as well. Ideas include harnessing the potential of big data and multi-modal sensing (Blasch et al. 2013), integrating sensor network with atmospheric dispersion models to simplify the processing (Ritter et al. 2016) and limited applications in construction. For instance, Akhavian and Behzadan (2015) created data-driven models for equipment activity recognition, and Vasenev et al. (2014) proposed a data collection framework for decision-making.

This review of literature shows that within the AEC/FM domain, current simulation methodologies do not facilitate the integration of sensor data into simulation components. In addition, due to the lack of universally accepted framework for the use of sensing and data collection technologies in AEC/FM seamless adoption of simulation-data integration protocols is hindered. The following Sub-section explores one of the chief barriers in this integration process: the inherent noise in sensor data.

II.1.2 Inherent noise of sensor data

In recent years, every aspect of the AEC/FM lifecycle has been examined as possible phases of improvement with advanced data sensing, computing technology, and information modeling (Golparvar-Fard et al. 2011; Leite et al. 2016). Functions such as improved project planning and delivery in construction, monitoring and control and establishing new industry standards and paradigm shifts for major decision-making have been explored by various researchers (Spencer Jr et al. 2004). For instance, work by Bathula et al. (2009) in transportation project monitoring, Chae et al. (2012) in structural health monitoring, Razavi and Hass (2010) in on-site material tracking, and Choe et al. (2014) in construction site safety have all demonstrated the versatile applications of these new technologies.

Given the complexity of tasks (i.e. multiple resources of different types) and the diversity of workforce (i.e. various trades each operating within their own physical spaces and constraints) involved in construction, producing a comprehensive picture of the project status requires incorporation of more than one type or class of sensors be deployed in form of a sensor network (a.k.a. grid) (Estrin et al. 2001). Khaleghi et al. (2013) defines this fusion as "the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision-making". This network comprises of a sensor (data collector), a processor (data handler), and a communication unit (data transmitter) in each node and the real-time data collected from

the project environment is transferred to central workstations for further analysis or as input to a decision support system. This process can culminate both on or off-site.

This proliferation of the type and quantity of sensors available is bound to produce an abundance of data, however, this may lead to a gap in data utilization as useful data is overlooked and contradictions are created in the interpretation based on end user's perception, expectations, or skillset and training. In addition, not all collected data is of expected quality and resolution. Further compounding this messy picture is the propensity towards low quality sensors due to high costs (procurement, installation, and maintenance costs) of high quality systems and significant noise in the data collected due to the harsh and dynamic environment of construction. Quality is further compromised in transmission of the collected data due to loss, latency, and reliability issues (Islam et al. 2012). Thus, the current status of sensor technology inhibits the collection of high quality data thus creating reliability issues in application. Zamalloa and Krishnamachari (2007) identified several factors that cause this variation and uncertainty in sensor reliability. This uncertainty is mainly a product of three related causes: human or machine error producing spurious readings, physical limitations of the sensors producing measurement errors such as approximation or truncation error, extraneous measurements collecting background data. These factors are further compounded by the stochastic nature of most data processing systems that increase the fuzziness in the data. (Colubi and González-Rodríguez 2015). These identified issues are considered as the chief barriers to adoption of new technology in the construction industry. since handling, cleaning, and post-

processing of raw sensor data requires special training and skills that are otherwise not expected from a trained construction engineer or project manager (Lee et al. 2013).

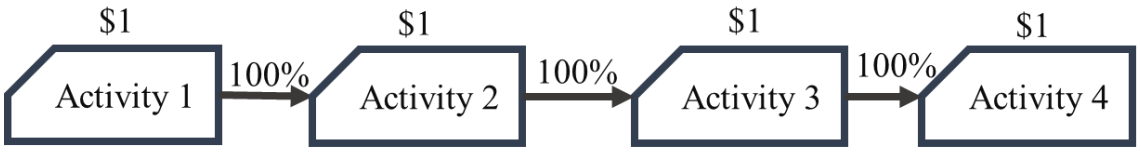
Further review of the literature also reveals another major obstacle to the widespread adoption of data capture technologies: transformation of raw data to information useful in decision-making requires significant processing. In this regard, Various data processing algorithms have been proposed as solutions. For example, Blasch et al. (2013) used data-driven simulation with applications in object tracking and traffic simulation to reduce uncertainty in data by analyzing trends in previously collected data. Hidden Markov models (HMMs) have been also used to detect anomalies in complex datasets (Flores et al. 2009) which are then removed or modified to reduce uncertainty. Some other algorithms dealing with noise have been proposed in general literature. For instance, Yang (2013) presented a decision tree algorithm to classify data into a hierarchical format, thus reducing the complexity of the data structure. However, the available solutions are mostly focused in dealing with continuous data streams and wide networks such as traffic systems, energy simulation, computer data streams, and anomaly in fluid flows. Processes to reduce data quality issues in discrete systems with defined start and end events are still limited as the discrete environment present unique challenges such as the great variation among the different instances of the same event. For instance, Ye et al. (2010) proposed a specification-based approach to identify isolated instances (with predetermined start and end times) of simple discrete human activities through matching extracted features with a standard vocabulary of previously extracted features. However,

this approach suffered from high volatility when dealing with uncertainty in the collected data.

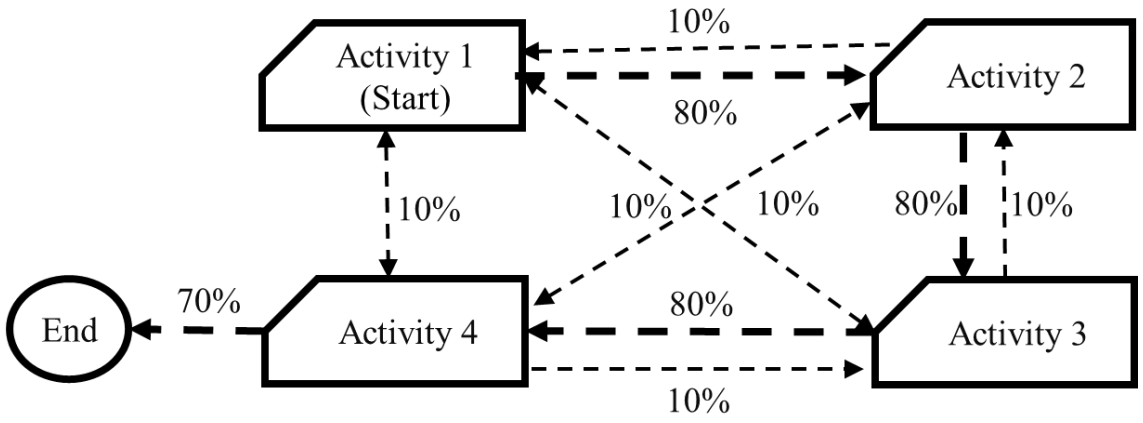
II.2 Chaos theory and imperfect sensor data

As previously stated, most of the data collected by sensors is not crisp and well differentiated: the collected data represent an imperfect manifestation of the real world with uncertain progressions and states (Izadi et al. 2015). This imperfection in the data available creates significant volatilities which can be explained using chaos theory. Chaos theory is the study of complex, nonlinear, dynamic systems (Lorenz 1963) that deals with systems that appear to be deterministic (e.g. a construction schedule) but can experience chaotic events (e.g. random variations). It illustrates mathematically that even deterministic systems can beave very unpredictably (i.e. chaotically). Thus, the dynamic interactions within result in hyper-sensitivity to the initial conditions overall. Lorenz (1963) expressed this succinctly as "the present determines the future, but the approximate present does not approximately determine the future". In common parlance, this phenomenon is widely known as the butterfly effect, a term first coined by Lorenz (1963), popularized by Gleick (1987), and later given a full mathematical treatment in the context of uncertainty in deterministic dynamic systems by Werndl (2009). With time, chaos theory has been expanded far beyond pure science (Levy 1994) with applications even in the social sciences (Kiel and Elliott 1996). Most application deal with dynamic systems that were beyond the theoretical frameworks available before chaos theory.

Within the scope of this Thesis, the implication of chaos theory is that if uncertain data from a sensor network is used to build a model of a dynamic construction system (even if the actual system appears linear and deterministic), the performance of the model can randomly change with a small change in initial conditions (i.e. accuracy of sensor readings). This proposition can be better explained using the activity networks in Figure II-1.



(a)



(b)

Figure II-1 Sample non-deterministic network

Figure II-1(a) represents a deterministic system where activity transitions are predetermined (no uncertainty). In contrast, Figure II-1(b) shows a non-deterministic (dynamic) system in which the transition from one activity to another is probabilistic. In this Figure, double arrows imply that the resource on a link can travel either way. It is worth noting that the activity network in Figure II-1(b) can be derived from the activity network in Figure II-1(a) essentially through introducing uncertainty in activity transitions. As a general rule, in order to derive the activity network in Figure II-1(b) from the one in Figure II-1(a), for any given Activity *i* (*i* = 1, 2, 3), the probability of the default succession (a.k.a. link strength value) is reduced by 20% (30%, for Activity 4), and a new arrow is added to connect Activity *i* to the remaining two activities (remaining two activities and the End node, for Activity 4). For instance, Activity 1 in Figure II-1(b) is 80% (rather than 100%) likely to be followed by Activity B, 10% likely to be followed by Activity 3, and 10% likely to be followed by Activity 4. Applying this rule to all activities in Figure II-1(a) generates the non-deterministic activity network in Figure II-1(b). Evidently, using the abovementioned succession alteration rule on only a subset of all activities can result in hybrid activity network which contain both deterministic and non-deterministic activity transitions.

Now, let's assume that the goal of each network is to move 100 items from Activity 1 to Activity 4. For simplicity, let's also assume that processing a single item in each activity in Figure II-1(a) and Figure II-1(b) costs $1. Since the activity network in Figure II-1(a) is predetermined, the total operation cost of moving 100 items in this network is

always equal to $400 (100 items multiplied by 4 activities at $1 per activity). Next the cost of moving 100 items in the network represented by Figure II-1(b) is examined and listed in Table II-1. In this Table, each iteration represents a hybrid activity network derived from Figure II-1(a) in which strength values (model input) of a random subset of links are altered by 10% within that subset. As results in Table II-1 indicate, even a slight alteration in the input creates a large volatility in the total operation cost (model output). For instance, the only difference between iterations 0 (benchmark) and 3 is that the strength value of the link connecting Activities 1 and 2 was changed from 1.0 in iteration 0 to 0.9 in iteration 1, resulting in an overall 5% change in the network strength values. However, this single alteration in the input results in a 25% increase in the output. Using elasticity terms, the cost is 5 times more elastic than the network strength values (i.e. a 1% change in network strength values changes the cost by 5%).

Table II-1 Volatility in network output due to change in input

Iteration	Overall change in network strength values	Total cost ($)	Overall change in cost
0	-	400	-
1	5%	500	25%
2	5%	470	18%
3	10%	558	40%
4	10%	556	39%
5	15%	798	100%

II.3 Research objective and contributions

As highlighted above, the combination of sensor data collection, processing imperfect data, and simulating heuristic systems is a nascent field with limited foundation but great potential. The body of work so far has been limited to isolated applications and lacks the specific knowledge required to transform the state of data-driven construction simulation modeling. The work presented here is motivated by the need to bridge these gaps by designing a scientific methodology, inspired by chaos theory and built upon an evolutionary algorithm, capable of refining imperfect (noisy) sensor data and generating clean datasets that can be used for simulation input modeling. The practical contribution of this work is that the output is not bound to the limitations in commercial sensing technology, thus allowing the use of low-cost sensors for data collection while minimizing the impact of inaccurate sensor readings on the overall quality of the simulation model. Ultimately, this approach is sought to promote simulation-based decision-making by reducing the upfront cost of data acquisition.

II.4 Methodology

In this Chapter, different steps of the designed methodology of refining imperfect sensor data for simulation input modeling are explained.

II.4.1 The design of the box moving experiment used to collect ergonomic sensor data

The experiment conducted in this research represents a warehouse operation in which workers transport boxes one by one from a loading area to an inspection area, inspect each box, and if the content is approved, move the box through the system to a designated unloading area. As shown in Figure II-2, the cyclic operation starts with a worker loading a box onto a cart and then pushing it to the inspection area. Next, an inspector lifts the box and inspects it. During inspection, the worker waits in the inspection area. After inspection, the inspector either accepts the box or rejects it. Upon acceptance, the worker lowers the box onto the cart, pushes it to the unloading area, unloads the box and then pulls the empty cart back to the loading area. If the box is rejected, however, the worker pulls back to the loading area with an empty cart. In both cases, the worker moves back to the loading area and the cycle starts over.

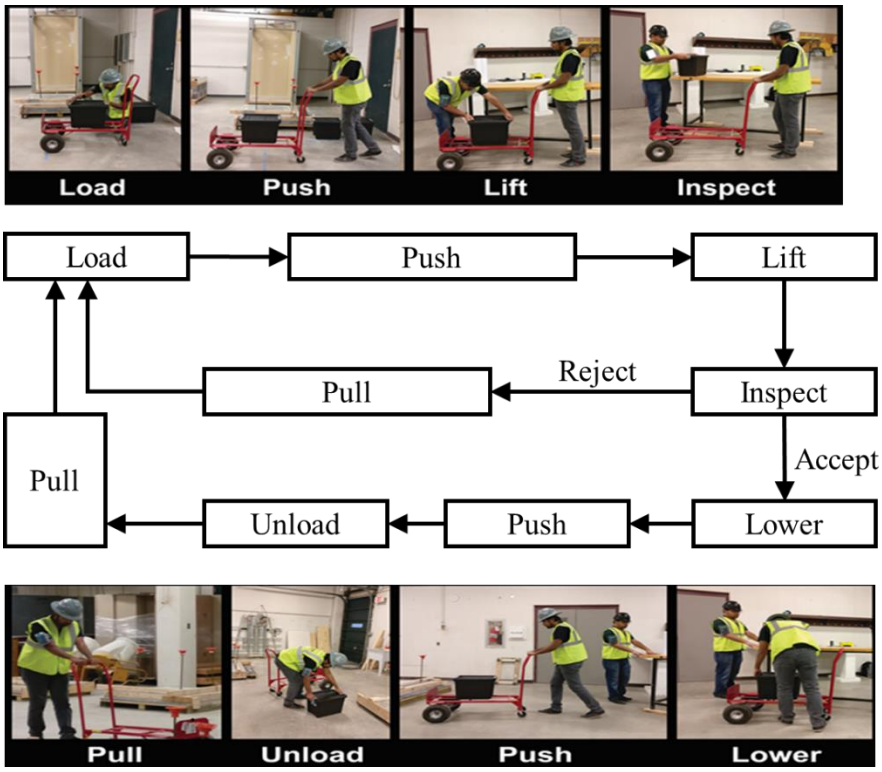


Figure II-2 Schematic workflow of the warehouse operation

This operation is performed for 15 cycles with worker W1 and inspector I1, and then repeated with worker W2 and inspector I2 for another 15 cycles. Two smartphones are mounted on each performer's body (one on upper arm and another on waist) for time-motion data collection.

II.4.2 Human activity recognition (HAR) algorithms to identify activities

In order to perform HAR, built-in sensors of each smartphone (accelerometer, linear acceleration, and gyroscope) are used to collect data at a frequency of 180 Hz with a 2-second window (Nath 2017). Here, accelerometer measures the acceleration force

including gravity, gyroscope measures the angular velocity and linear accelerometer measures the acceleration force excluding gravity. Accelerometer and gyroscope are hardware sensors, whereas linear accelerometer is a software sensor. Details of the data collection and preparation process are summarized in Table II-2.

Table II-2 Summary of the data preparation process

Category	Summary
Collected Sensor Data	Accelerometer (X, Y, Z), Linear-Accelerometer (X, Y, Z), Gyroscope (X, Y, Z)
Extracted Sensor Data	Accelerometer-Jerk (X, Y, Z), Linear-Accelerometer-Jerk (X, Y, Z), Gyroscope-Jerk (X, Y, Z), Accelerometer-Magnitude, Linear-Accelerometer-Magnitude, Gyroscope-Magnitude, Accelerometer-Jerk-Magnitude, Linear-Accelerometer-Jerk-Magnitude, and Gyroscope-Jerk-Magnitude.
Sampling Rate	180Hz after processed into time series of uniform interval.
Window Size	360 data points (2 seconds)
Statistical Features	Mean, Maximum, Minimum, Standard Deviation, Mean-Absolute Deviation, Interquartile Range, Skewness, Kurtosis, Autoregressive Coefficients.
No. of Extracted Features	576
No. of Selected Features	125 for Worker, 84 for Inspector
Feature Selection Algorithm	ReliefF
Classifier Algorithm	Multi-class Support Vector Machine

Following data collection, a series of machine learning algorithms is used to transform pure sensor data to discrete activity sequences. This pre-processing phase

produces activity sequence duration information for each worker and inspector. Given the presence of noise and errors in collected data and HAR algorithm implementation, a post-processing stage is necessary to further improve the accuracy and consistency of the resulting information.

In general, three main sources of error exist in the collected data. First, sensor under-sampling (freezing) where a sensor stops working for a few seconds causing gaps in data. Second, sensor oversampling which normally occurs after a period of sensor freezing, and causes the sensor to collect data at a faster rate to compensate for the missing data points during the freezing period, thus creating redundancy in collected data. These first two errors are normally compensated using linear interpolation, and by removing redundant data points. The third type of error, unlike the other two is human error which occurs when the person from whom training data is collected for HAR, performs activities other than those planned, thus creating subsets in training data that cannot be correctly classified (Nath 2017). Resolving this error is more complicated since training subjects behave differently, and there is no single formula that can handle all such erroneous instances (Akhavian et al. 2015). It is imperative that the presence of these systematic and human errors impact the accuracy of HAR. For instance, as reported in Nath (2017), while some activities are recognized with good accuracy (ranging from low 80% to 99%), there is still significant confusion between specific activities (e.g. ‘load’ and ‘unload’) which reduced the overall fidelity of HAR process.

II.4.3 Simulation input modeling

The activity sequence identified by the HAR algorithm is used to generate an activity transition matrix, hereinafter referred to as the dependency network assimilator (DNA). Elements of this matrix help identify the sequence of activities as occurred in the real system and captured by sensor data. However, it must be noted that according to chaos theory, the activity level discrepancies between the actual and identified activity sequences, if used as input in a larger system such as a simulation model representing complex and dynamic environments, the model would quickly accrue significant inaccuracies in output (e.g. completion time, projected cost, productivity). Thus, in order to maintain the reliability of the system, it is important to minimize the error.

The discrepancy between ideal and extracted DNA matrices is illustrated through the example presented in Figure II-3 which shows sample observed (ground truth) and extracted (imperfect) DNA matrices for a project consisting of three activities. The observed DNA matrix of Figure II-3(a) shows that Activity *X* is proceeded three times by Activity *Y*, and eight times by Activity *Z*. Similarly, Activity *Y* is followed five times by Activity *Z* and nine times by Activity *X*. Finally, Activity *Z* is proceeded two times by Activity *X* and eleven times by Activity *Y*. In comparison, Figure II-3(b) shows that the extracted DNA matrix of the same project, obtained from the output of HAR using raw sensor data, contains erroneous activity sequences. Such errors can be attributed to inherent inaccuracies in sensor readings and the limitations of the HAR algorithm in correctly identifying activities from sensor data.

		Succeeding activity		
		X	*Y*	*Z*
Proceeding activity	*X*	0	3	8
	Y	9	0	5
	Z	2	11	0

(a)

		Succeeding activity		
		X	*Y*	*Z*
Proceeding activity	*X*	0	11	0
	Y	3	2	13
	Z	8	5	3

(b)

Figure II-3 Activity sequence matrix (DNA) as described by (a) ground truth, and (b) extracted HAR information

For example, the extracted DNA matrix identifies two instances of *Y*-*Y* transition and three instances of *Z*-*Z* transition, which are all incorrect. Moreover, in certain cases, while correct activity transitions are detected, the number of such transitions is not correctly identified. For example, according to the extracted DNA matrix, 11 *X*-*Y* transitions (instead of 3) and 8 *Z*-*X* transitions (instead of 2) are detected. This example makes it clear that the final results obtained from a simulation model will vary based on whether the ideal or extracted DNA matrix is used as the basis of the activity cycle diagram (ACD) and the corresponding simulation model. While it is preferable to use the ideal DNA matrix, this matrix can be extracted only under perfect conditions with sensor data that is 100% accurate, and through the use of highly-trained (error-free) HAR algorithms.

However, achieving 100% accuracy is almost impossible (for reasons discussed above). Therefore, the challenge is to use the extracted DNA matrix (with intrinsic fuzziness) to create a simulation model that can still closely mimic the real system and predict its performance with high fidelity. In this Chapter, an evolutionary approach combined with simulations is proposed and tested to achieve this goal. Figure II-4 shows the main building blocks of the designed methodology.

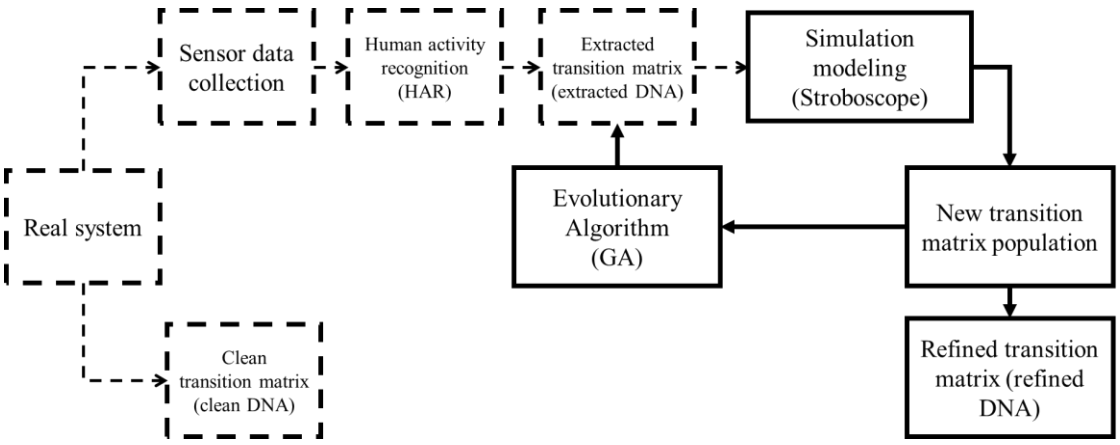


Figure II-4 Block diagram of designed sensor data refinement methodology

This block diagram illustrates the path of progression of the algorithm from the initial data collection to initial activity recognition through HAR and the use of the HAR results to create a probabilistic ACD of the operation. This representative ACD is implemented as a DES model in Stroboscope. Stroboscope is a programmable and extensible simulation authoring system designed for modeling complex construction operations (Martinez 1996). Several iterations of the model are run and given the

probabilistic (uncertain) nature of the ACD, it is expected that each iteration results in a new (and slightly different) DNA matrix (a.k.a. children population in genetic algorithm or GA). This new pool of DNA matrices is subsequently used in an evolutionary process to generate a cleaner DNA matrix. In each step, the generated DNA matrix undergoes fitness evaluation, and then fed to the DES model. This process repeats until results converge. The combination of simulation and GA enables the production of a refined DNA matrix from the noisy sensor data. The following Sub-sections contain detailed discussions about this process.

II.4.4 The deterministic simulation model

As illustrated earlier, the warehouse operation experiment consists of independent activities, each with discrete start and end times. These activities can be defined as separate nodes in a DES network connected by links carrying resources (i.e. worker, inspector, boxes) which are defined and stored in queues.

II.4.4.1 DES model with clean activity transitions

The ACD shown in Figure II-5 illustrates the deterministic DES model of the warehouse operation experiment with clean (ideal) transitions between successive activities. This model is validated through a point-by-point comparison with the video recording of the real experiment at random times, thus ensuring that it was an accurate representation of the real system in terms of the operation logic and activity durations. The correct validation of this model also guarantees that transitions between successive

activities are deterministic (non-probabilistic), thus yielding an ideal DNA matrix. As previously described, each element in the DNA matrix represents the strength (i.e. likelihood) of transitioning from a preceding activity to a succeeding activity.

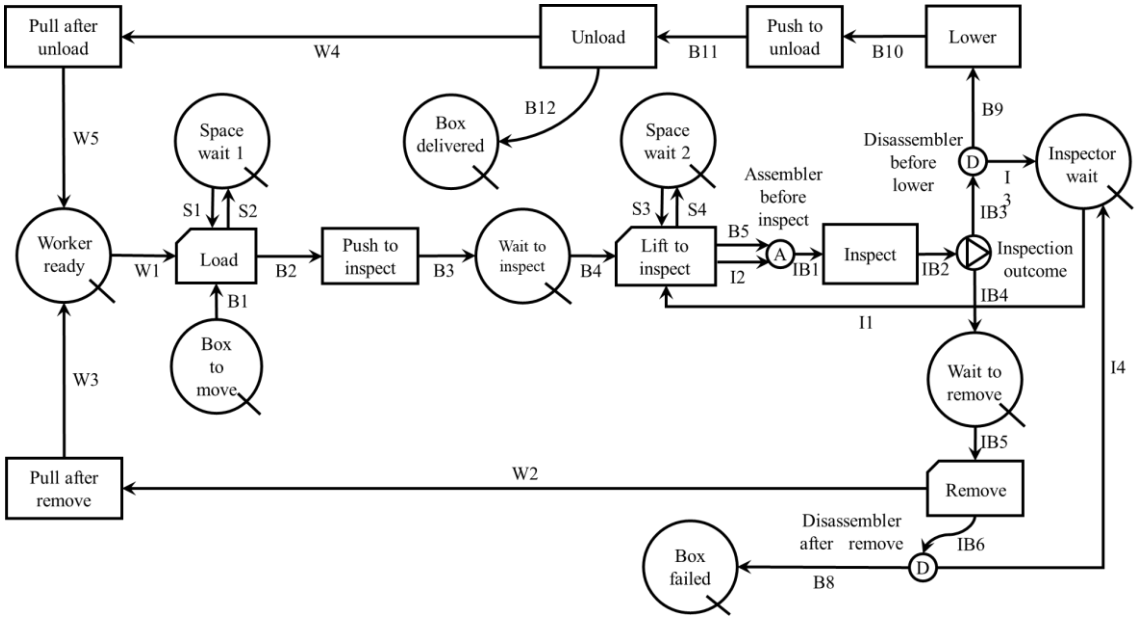


Figure II-5 ACD Diagram of the cyclic warehouse operation

It should be noted that while most activities shown in Figure II-5 qualify as both preceding and succeeding activities, some are only of one type; for instance, Activity ‘load’ is only a preceding activity as it starts a cycle, whereas Activities ‘unload’ and ‘remove’ are only succeeding activities as they end the cycle. Therefore, the DNA matrix does not contain an equal number of preceding and succeeding activities, and consequently may not necessarily be a square matrix. This clean DNA matrix, as shown

in Figure II-6 shows ideal transitions between activities in each sequence. Rows represent preceding activities and are sequentially numbered (i = 1 to n), whereas columns represent succeeding activities and are sequentially numbered (j = 1 to m). As expected, almost all rows hold binary (single non-zero) values since each activity is only followed by one succeeding activity. The only exception to this rule is Activity ‘inspect’, which depending on the outcome of the inspection, can be followed by either Activity ‘lower’ or Activity ‘reject’.

Load	Push to inspect	Lift to inspect	Inspect	Lower	Remove	Push to unload	Unload	Pull after unload	Pull after remove	Ready to load	Box delivered	
0	30	0	0	0	0	0	0	0	0	0	0	Load
0	0	29	0	0	0	0	0	0	1	0	0	Push to inspect
0	0	0	28	0	0	0	0	1	0	0	0	Lift to inspect
0	0	0	0	20	10	0	0	0	0	0	0	Inspect
0	0	0	0	0	0	18	0	0	0	0	0	Lower
0	0	0	0	0	0	0	0	0	0	10	0	Remove
0	0	0	0	0	0	0	18	0	0	0	0	Push to unload
0	0	0	0	0	0	0	0	0	0	0	20	Unload
0	2	0	0	0	0	0	1	0	0	0	0	Pull after unload
0	2	0	0	0	0	0	1	0	0	0	0	Pull after remove
30	0	0	0	0	0	0	0	0	0	0	0	Ready to load

Figure II-6 Clean (ideal) DNA matrix (ground truth) of the warehouse operation

II.4.4.2 Activity duration modeling in Stroboscope

The DES model requires a specified activity duration distribution in order to run realistic simulation. Thus, experimental results are used to extrapolate the activity duration distribution for each of the activities in order to build a representative model. As enumerated by to Law et al. (1991), experimental data can be used in three ways in a simulation: selecting one of the observed data points every time, randomly using a sample from collected model, and fitting a theoretical data to the model. The first two methods have been invalidated by previous research as an ineffective input method to build a dynamic simulation model (Akhavian 2015), thus, in order to incorporate the rage and variability of the dataset the third method is chosen for this implementation.. Stroboscope can model Scaled Beta, Erlang, exponential, Gamma, Normal, PERT Beta, triangular, and uniform distributions (Martinez and Ioannou 1994). In this research, these distributions are tested for goodness of fit in describing extracted activity durations using three tests: Chi-Square, Kolmogorov-Smirnov (K-S), and Anderson-Darling (A-D) (Banks 1998). Table II-3 shows the results of the goodness-of-fit tests, their rankings, and the numerical total of the ranks for Activity ‘unload’. Since the Normal distribution results in the best total ranking, it is ultimately selected to describe the duration of Activity ‘Unload’ in the simulation model. Similar analyses are conducted for all other activities.

Table II-3 Ranking of the best fitted probability for Activity ‘Unload’

Distribution	K-S		A-D		Chi-Squared		Sum of Ranks
	Statistic	Rank	Statistic	Rank	Statistic	Rank	
Beta	0.28296	2	11.422	8	N/A		10
Erlang	0.37122	6	2.6008	3	0.31304	3	12
Exponential	0.55156	8	6.1967	5	1.7434	6	19
Gamma	0.32795	4	2.1805	2	0.20342	1	7
Normal	0.30586	3	1.6871	1	0.219	2	6
PERT	0.27188	1	9.3159	7	0.38599	4	12
Triangular	0.42002	7	6.964	6	1.2962	5	18
Uniform	0.34273	5	5.6715	4	N/A		9

The selected distribution and its parameters for each activity is shown in Table II-4. In addition, classification results are used to determine the probability of a box accepted or rejected. For instance, it is found that 29 instances of Activity ‘load’ followed Activity ‘pull’ which implies that in total, 29 boxes are moved in the system. Similarly, 20 instances of Activity ‘unload’ followed Activity ‘push’ which means that 20 boxes are accepted by the Inspector. Therefore, it can be inferred that the ratio of accept/reject is 20:9.

Table II-4 Selected distributions and their parameters for activity durations

	Activity	Distribution	Parameters
Worker	Load	Gamma	a = 22.57 b = 0.28418
	Unload	Scaled Beta	Low = -29.258 High = 21.09 $\alpha 1$ = 196.84 $\alpha 2$ = 92.451
	Lower	Normal	μ = 1.3086 σ = 6.4737
	Push to Inspect	Gamma	a = 120.05 b = 0.08776
	Push to Unload	Uniform	Low = 11.709 High = 19.291
	Pull after Reject	Normal	μ = 3.7786 σ = 12.5
	Pull after Unload	Normal	μ = 6.3539 σ = 28.091
Inspector	Lift	Normal	μ = 0.91676 σ = 3.0741
	Inspect	Normal	μ = 4.4341 σ = 14.375
	Reject	Uniform	Low = 1.6515 High = 3.5487

II.4.4.3 Model validation

The next step in the implementation workflow is to test the validity of the model build as a representation of the experiment. In particular, the robustness, the scalability and activity level accuracy is tested. In Simulation 1, the scalability of the model is tested by running the model 30 times with 1 worker and 1 inspector moving 30 to 900 boxes. Results in terms of ratio of expected (from real system) and obtained (from simulation)

total time, inspector's idle time, and worker's idle time are shown in Figure II-7. This Figure shows that the obtained time is within 10% of the experimental results for all three parameters thus validating the scalability of the model.

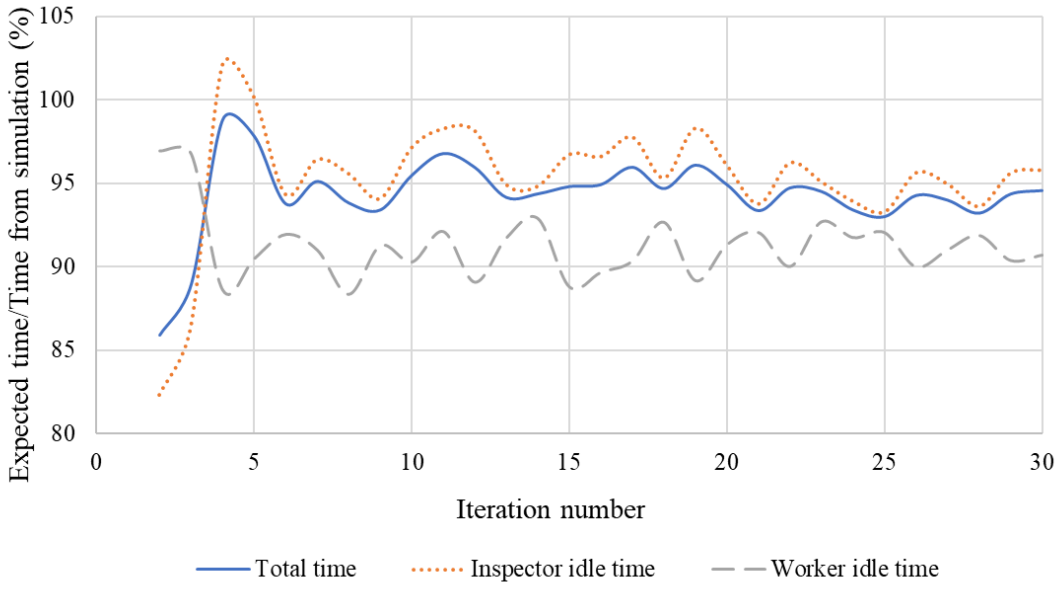


Figure II-7 Ratio of expected and simulation times for simulation 1

The robustness of the model is tested in simulation 2 by running the model 1000 times with 30 boxes. Figure II-8 shows that in terms of the total time of the operation and idle time of the inspector, on average, simulation results are 6% lower than experimental results. This can be attributed to the seamless transition between simulated activities unlike in the real system where transitions take time. Furthermore, the activity recognition

algorithm removes false detections (FDs) from the data, thus contributing to the slight difference by weeding out the extreme values.

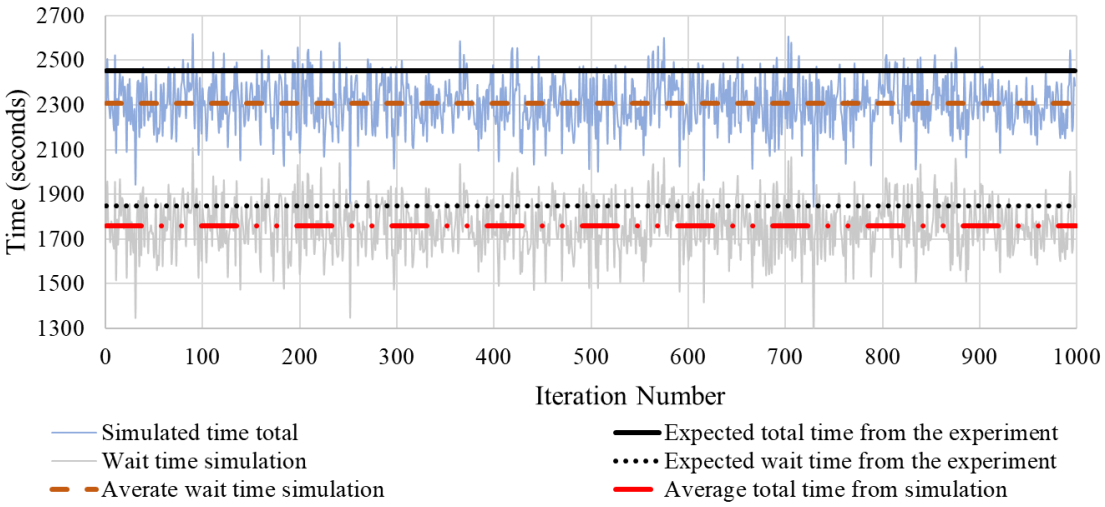


Figure II-8 Simulation robustness in estimating total time and inspector's idle time

Finally, the validity of the activity duration is examined in simulation 3 by moving 1,000 boxes. The ratios of the average activity durations between the real world and simulation results are computed and shown in the radar chart of Figure II-9. As seen in this Figure, while the ratio of the durations from simulation model and durations from HAR is the most accurate of the three ratios, which highlights the validity of the simulation model, the least accurate ratio is the ratio of durations from HAR and observed durations, suggesting deficiencies in the quality of collected sensor data. Also, the ratio of durations

from the simulation model and the observed durations is on average 90%, which is a decent approximation of the real world by the developed simulation model.

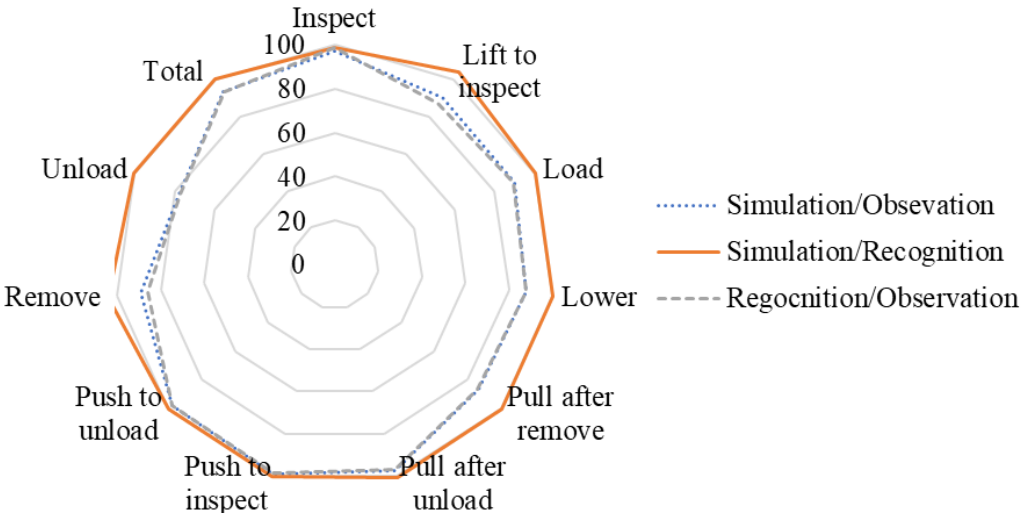


Figure II-9 Simulation robustness in estimating activity durations

II.4.5 The non-deterministic simulation model

II.4.5.1 DES model with probabilistic activity transitions

As previously discussed, the output of the HAR algorithm contains noise due to imperfect sensor data and/or inaccuracies in the HAR algorithm. Thus, in the ACD diagram generated using this output, each activity can be followed by a number of other activities even though that might not be the case in the real system. Hence, the DES model built on this dataset is probabilistic (non-deterministic) where each activity can be followed by any other activity. A hypothetical scenario showing a non-deterministic ACD

with four activities was illustrated in Figure II-1(b). Similarly, the precedence logic obtained for the warehouse operation experiment results in the extracted DNA matrix of Figure II-10. In contrast to the clean DNA matrix in Figure II-6, some rows in the extracted DNA matrix contain multiple non-zero values indicating varying degrees of noise. For instance, the extracted DNA matrix includes multiple transitions from Activity ‘load’ to Activity ‘unload’, Activity ‘lower’ to Activity ‘lift to inspect’, or Activity ‘push to unload’ to Activity ‘pull after unload’. However, neither of these transitions did occur in the real system, indicating that these and similar elements in the extracted DNA matrix have resulted from error propagation through sensor data collection and HAR algorithm.

Load	Push to inspect	Lift to inspect	Inspect	Lower	Remove	Push to unload	Unload	Pull after unload	Pull after remove	Ready to load	Box delivered	
0	26	2	0	0	0	0	2	0	0	0	0	Load
2	0	23	0	0	0	0	0	2	2	0	0	Push to inspect
0	0	0	25	0	0	0	3	2	0	0	0	Lift to inspect
0	0	0	0	20	10	0	0	0	0	0	0	Inspect
0	0	2	1	0	0	15	1	1	0	0	0	Lower
0	0	0	0	0	0	0	0	0	0	10	0	Remove
0	0	0	0	0	0	0	16	2	2	0	0	Push to unload
1	0	3	4	0	0	4	0	0	0	0	20	Unload
0	2	0	0	0	0	1	4	0	0	0	0	Pull after unload
0	2	0	0	0	0	0	2	0	0	0	0	Pull after remove
30	0	0	0	0	0	0	0	0	0	0	0	Ready to load

Figure II-10 Extracted (noisy) DNA matrix of the warehouse operation

II.4.5.2 Modeling probabilistic activity transitions in Stroboscope

Obtaining reliable results from simulation requires that the real system be modeled with sufficient accuracy and fidelity. In order to model the uncertainties in the precedence logic (such as those shown in the extracted DNA matrix of Figure II-10), the deterministic DES model needs to be expanded with new capabilities. For this reason, a standard modeling element called fork is added to each of the Activities in the DES model. In Stroboscope, fork elements are probabilistic elements that connect an activity with several other Activities with links having numerical strength (weight) values. During execution, an outgoing link is picked on a random basis by considering the designated relative strength values of the outgoing links (Martinez 1996). In the context of the warehouse operation experiment, strength values were defined using values from the extracted DNA matrix of Figure II-10. This enabled the DES model to allow multiple outgoing links from each activity, thus resembling the fuzzy behavior.

To implement this fuzzy DES model, and given the specific syntax of Stroboscope, three types of Activities with different implementation mechanisms must be defined: initiation activity, simple activity, and termination activity. An initiation activity is used to start a new cycle (e.g. box moving cycle), a termination activity is used to end a cycle, and a simple activity is used in all other cases. Figure II-11 shows a partial ACD diagram in which these activities are implemented. The cycle starts with initiation Activity 1, proceeds to simple Activity 1, continues onto simple Activities 2, 3, or 4, or ends in termination Activity 1 (on a random basis), which then closes the cycle and releases

resources back to initiation Activity 1. As a convention, a solid link represents deterministic flow and a dotted link shows probabilistic (fuzzy) flow. To comply with Stroboscope syntax, as illustrated in Figure II-11, an initiation activity has one or more preceding queues that feed in resources, is proceeded by a fork with a single outgoing link to a queue, and ultimately connected to another Activity in the network. The subsequent activity (e.g. simple Activity 1 in Figure II-11) is then followed by a fork that provides necessary connections to successive Activities (through queues). As previously mentioned, the selection of the outgoing link from this fork is random and is based on the strength values of all outgoing links from that fork. Ultimately, when a resource arrives at a termination activity, it is simply forwarded to the following queue causing the cycle to close. In case a resource needs to be regenerated at the closure of a cycle, an action event can be invoked in Stroboscope. This is shown as the bold dashed line outgoing from termination Activity 1 to the 'worker ready' queue in Figure II-11. More details about action events and resource generation are beyond the scope of this Chapter and can be found in Martinez (1996). In particular, the non-deterministic model created in Stroboscope to represent the warehouse operation experiment contains 1 initiation activity (modeling the beginning of the loading cycle at Activity 'load'), 7 simple Activities (modeling Activities 'push to inspect', 'lift to inspect', 'inspect', 'lower', 'push to unload', 'pull after unload', and 'pull after remove'), and 2 termination Activities (modeling the end of the cycle at Activities 'remove' and 'unload').

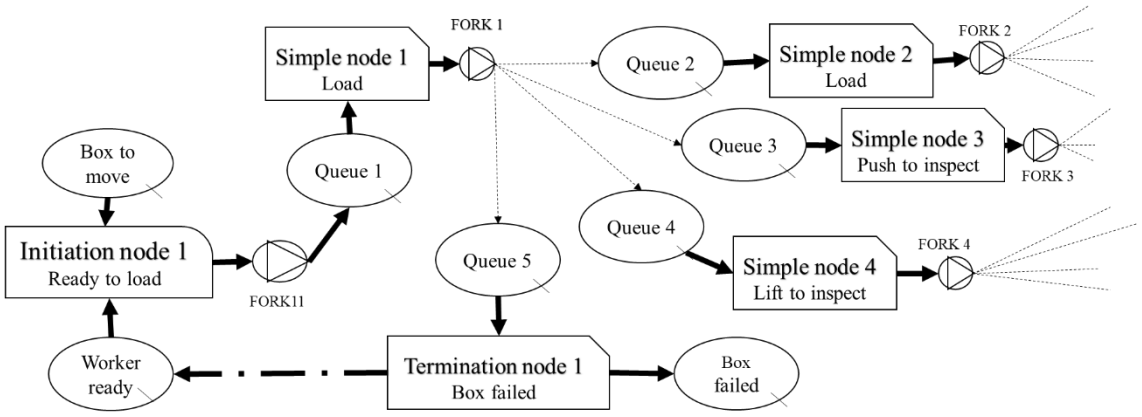


Figure II-11 Partial fuzzy ACD diagram illustrating different activity types

II.4.6 Refining the extracted activity transition matrix

II.4.6.1 Implementation of evolutionary algorithm

A new evolutionary GA-Based technique is designed and implemented to reduce the errors in the extracted DNA matrix and transform it to a refined (close to ideal) DNA matrix. GA has been applied extensively in a wide range of fields including water contamination characterization (Preis and Ostfeld 2008), evaluating construction plans using data environment analysis (Torabi and Mahlooji 2017), site layout planning for construction projects (RazaviAlavi and AbouRizk 2016), and speech recognition based on random projections (Kataoka et al. 2016). In general, a GA-implementation is based on five key operations to iteratively improve the solution and eventually reach an optimal solution from a number of possible (not optimal) solutions (Poli et al. 2008). In this research, these five principles are implemented to refine the extracted DNA matrix, as shown in Figure II-12 and briefly described in the following paragraphs.

- Stage 1 – Define the mother species: The extracted DNA matrix generated by the HAR algorithm is designated as the initial mother species in the implementation. The value of each element in the mother matrix is taken as the strength value of the link and is represented by $\gamma_{(ij)}$, where *i* is the row index and *j* is the column index. This matrix is thus used to generate the first generation of daughter matrices.
- Stage 2 – Create population of daughters: As discussed previously, the non-deterministic DES model is built and run several times, each producing a new daughter matrix. In each iteration, forks are evaluated given the strength values of their outgoing links. This results in anomalies in activity transitions leading to a population of daughter DNA matrices with inherent uncertainty. This intentional uncertainty helps create the population of daughter matrices and perfectly represents the natural uncertainty in transitions.
- Stage 3 – Evaluate fitness: In this stage, the fitness value of the each of the daughter DNA matrices are assessed by predefined fitness criteria. If the fitness value of a daughter matrix meets the criteria of acceptance, it will be chosen as the final matrix. In terms of GA workflow, this is termed the stopping condition.
- Stage 4 – Create mating pool: The daughter matrices are ranked based on the value of their fitness parameter ω, and a subset of the available daughter matrices is selected to generate the next group of mother matrices. This subset is also known as the mating pool of daughter matrices.

- Stage 5 – Produce a new generation: Once the mating pool is selected, a new generation of mother DNA matrices are created by implementing crossover, elitism, and mutation on the daughter matrices of the mating pool (Davis 1991; Reeves 2003). Crossover combines parts of two or more daughter matrices, mutation changes random parts of certain daughter matrices, and elitism simply carries on daughter matrices that meet certain criteria to the next generation (Srinivas and Patnaik 1994).

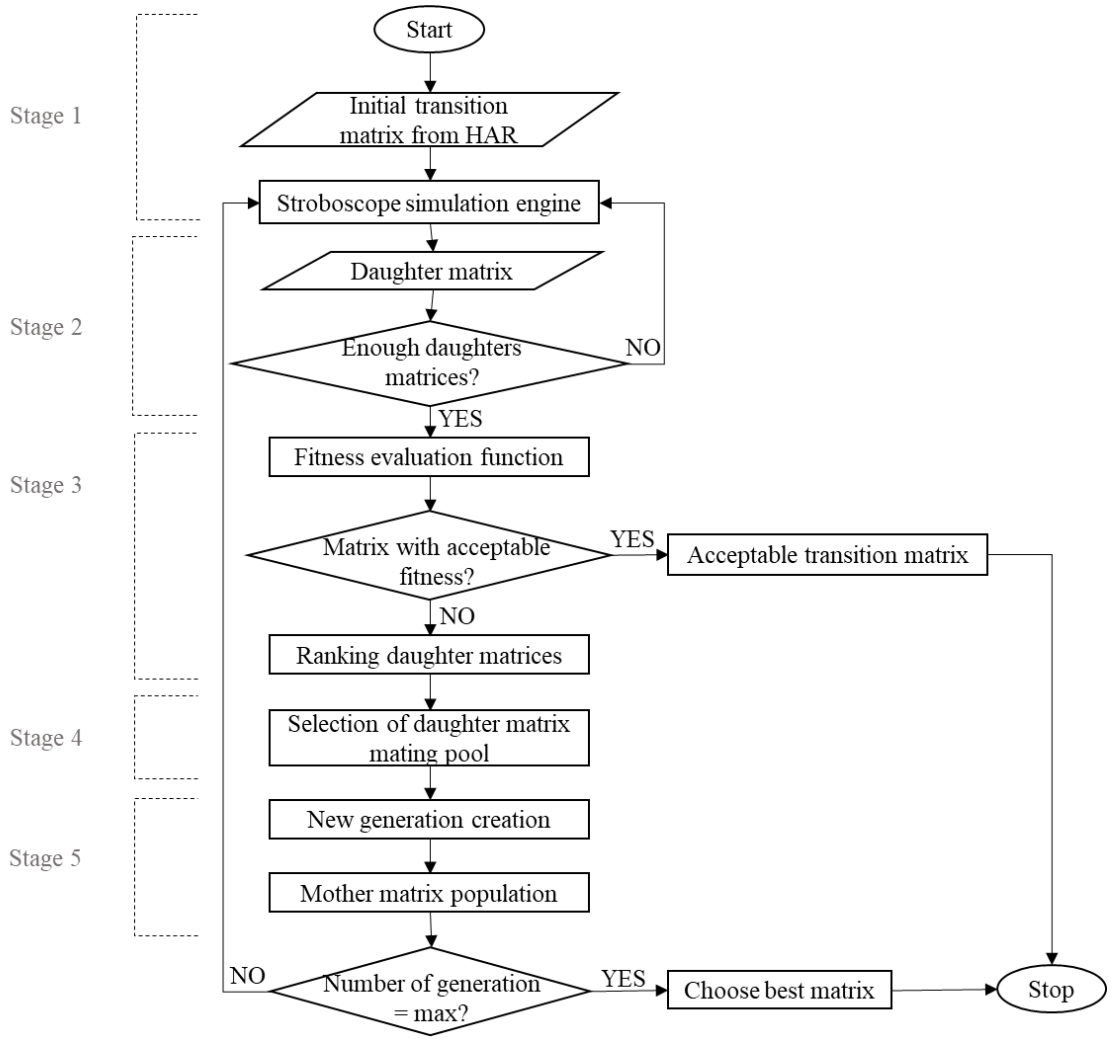


Figure II-12 GA workflow to refine the extracted DNA matrix

II.4.6.2 Fitness function

The fitness function is unique to each GA implementation. In the context of the warehouse operation experiment, this function is formulated based on the expected relationship between initial (extracted) and final (refined) DNA matrices. In particular, several field observations are made to reduce the complexity of the invoked GA functions,

and help translate and preserve the physical constraints in the intermediate transition matrices generated by the GA. These logical rules are referred to as hard constraints (Chan et al. 1996; Yu and Buyya 2006), and are listed below:

- Rows in the ideal (clean) DNA matrix are binary (only one non-zero element in each row) except where a determination is to be made as to where to move a resource after a decision activity (e.g. inspector station). In that case, there may be more than one non-zero element in a single row. Thus, an overall binary matrix was taken as the final goal of the experiment. For exceptions to this rule, see the note below.
- The activities in which a decision is to be made is called a chance node. In each stage of the GA implementation, strength values of the outgoing links from a chance node are assumed to be known. For instance, in the warehouse operation experiment, Activities 'inspect' is classified as a chance node; here, the inspector makes a decision on whether to accept or reject a box. Thus, in the row of the DNA matrix corresponding to this node, the number of accepted vs. rejected boxes (as observed in the experiment and recognized by the HAR algorithm) were inserted as non-zero elements. In particular, the HAR algorithm identified 20 instances of Activity 'lower' (conducted by the worker immediately after the box was approved) and 10 instances of Activity 'reject' (conducted by the inspector immediately after the box was rejected). Thus, 20 and 10 were used in the corresponding row of the extracted DNA matrix.

- Once a resource completes an activity, it moves to the next activity. In GA implementation, this translates into the rule that once a resource leaves an activity, it does not immediately return to that activity. As a practical matter, this is an acceptable assumption since having a short loop (a loop starting and ending in the same activity) is not likely to happen in an ACD. This logical observation can be used to infer that in a DNA matrix, diagonal elements must be zero.

The mathematical boundaries governing the processing of transition matrices during the GA are based on the hard constraints discussed above. Moreover, it is assumed that despite the presence of noise in individual data, in the context of large datasets, the data collected by sensors and processed through HAR algorithm is reasonably reliable. This foundational assumption is used in the analysis of the extracted DNA matrix where the strongest links (large non-zero values in the matrix) are assumed to be more likely to be statistically reliable, and thus should to the most extent preserved (and not utterly diminished) during the GA implementation. Considering these arguments and given the hard constraints described above, $\gamma(i)$ is defined as the strength value of row *i* in a daughter matrix, and $\gamma(ij)$ as the strength value of a particular transition from Activity *i* to Activity *j* in a daughter matrix. Equation II-1 shows the formulation of the fitness parameter of each row based on these principles. This parameter is defined as the ratio of the maximum strength value in that row to the sum of all strength values in the same row.

$$\omega_d(i) = \frac{\max(\gamma(i))}{\sum_{j=1}^{n} \gamma(ij)} \qquad \text{(II-1)}$$

Additionally, the overall fitness parameter of the matrix, shown in Equation II-2 is calculated as the arithmetic average of the fitness parameters of all rows.

$$\omega_d = \bigvee_{i=1}^{n} \widehat{\omega_d(i)} \tag{II-2}$$

Next, the objective function of the GA is defined as the maximization of this overall fitness parameter. In this work, an average value of 0.95 is selected as a good benchmark. This is represented mathematically by Equation II-3.

$$Z = \max \omega_d \tag{II-3}$$

Moreover, owing to constraints of time in practical applications and the ultimate goal of near-instantaneous updating of the model, the number of generations can be altered to maintain processing efficiency. For the warehouse operation experiment, 10 generations are deemed to be sufficient to produce stable results. These constraints define the stopping condition as shown in Equation II-4.

$$Z > 0.95 \text{ OR generation number} = 10 \tag{II-4}$$

II.4.6.3 Parameters of the GA

The parameters used to produce a feasible solution mainly depend on the quality of input data, expected accuracy of the final results, available computation time, and processor quality. Considering these criteria, the following parameters are chosen for the warehouse operation experiment discussed in this paper:

- No. of mothers in each generation: 3
- No. of daughters generated by each mother: 5

- No. of daughters in each generation: 15
- No. of generations: 10
- Acceptable parameter of fitness: 0.95

The number of mother and daughter matrices, as well as the number of generations are directly proportional to the complexity of the GA implementation, and the time budgeted for processing. Thus, increasing either of the parameters can improve overall accuracy. Also, the number of generations is often the best way to control the overall computation time. Finally, since it is not possible to achieve the ideal (clean) DNA matrix, an acceptable parameter of fitness is specified to select the best possible refined DNA matrix that resembles the clean DNA matrix to the most extent possible. Once this fitness is achieved, the GA implementation is terminated. It must be noted that the criteria and the values specified here are a result of a mainly qualitative process and are thus expected to vary depending on the application and context.

II.5 Results and analysis

II.5.1 Evaluating the effectiveness of GA implementation

The extracted DNA matrix shown in Figure II-10, is used as the initial mother matrix in the implementation of the developed GA-Stroboscope model. Initially, the model is run with the same initial mother matrix 3 times to obtain the first generation of mother matrices and henceforth, the model is launched 5 times for each of the mother matrices to obtain the 15 daughter matrices. After evaluation and selection, this process is

repeated 10 times representing the 10 generations. Thus, the refined DNA matrix shown in Figure II-13 is obtained upon termination of the process. The refined DNA matrix is observed to resemble the clean DNA matrix in Figure II-6 more closely than the extracted DNA matrix of Figure II-7. For instances, the 17 erroneous transitions from the extracted DNA matrix have been treated and reduced to only four fuzzy transitions (from Activities 'pull after remove' and 'pull after unload' to Activity 'unload', from Activity 'push to inspect' to Activity 'pull after remove', and from Activity 'lift to inspect' to Activity 'pull after remove'). Moreover, the extracted DNA matrix contained only 4 perfectly binary columns which was increased to 11 out of 12 in the refined DNA matrix.

Load	Push to inspect	Lift to inspect	Inspect	Lower	Remove	Push to unload	Unload	Pull after unload	Pull after remove	Ready to load	Box delivered	
0	30	0	0	0	0	0	0	0	0	0	0	Load
0	0	29	0	0	0	0	0	0	1	0	0	Push to inspect
0	0	0	28	0	0	0	0	1	0	0	0	Lift to inspect
0	0	0	0	20	10	0	0	0	0	0	0	Inspect
0	0	0	0	0	0	18	0	0	0	0	0	Lower
0	0	0	0	0	0	0	0	0	0	10	0	Remove
0	0	0	0	0	0	0	18	0	0	0	0	Push to unload
0	0	0	0	0	0	0	0	0	0	0	20	Unload
0	0	0	0	0	0	0	1	0	0	0	0	Pull after unload
0	0	0	0	0	0	0	1	0	0	0	0	Pull after remove
30	0	0	0	0	0	0	0	0	0	0	0	Ready to load

Figure II-13 Refined (final) DNA matrix of the warehouse operation

As previously stated, the overall fitness of the entire matrix (representing the binary nature of the matrix) is a key evaluation parameter of matrices in each generation. In essence, this parameter provides an indication of the percentage of correct transitions. In the extracted DNA matrix, this value was only 0.74 whereas in the refined DNA matrix it increased by 30% to 0.96 (as shown in Figure II-14), which is sufficiently close to the clean (ideal) DNA matrix fitness parameter of 0.97. Another indicator of the effectiveness of the GA implementation is that the average of the fitness parameter steadily increases with each new generation of daughter matrices. Figure II-14 demonstrates that each iteration improves the fitness of the transition matrix.

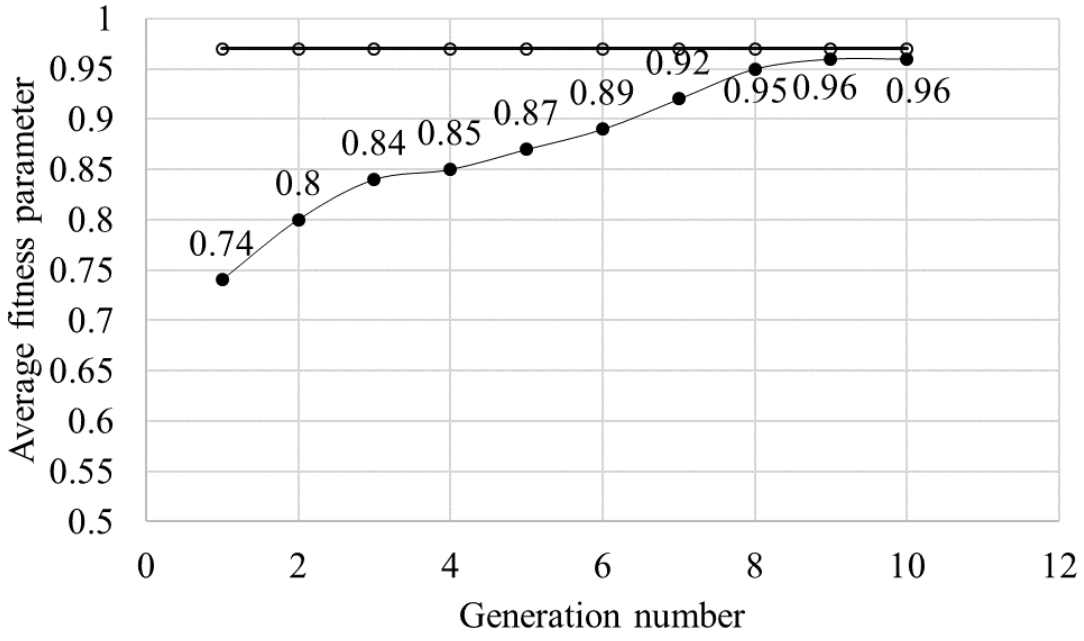


Figure II-14 Average fitness parameter of the resulting DNA matrix after each generation

II.5.2 Investigating the impact of DNA refinement on DES results

Overall, the ultimate goal of implementing the GA-Stroboscope model to refine the imperfect sensor data is to produce a more reliable input for simulation models of the experiment. Thus, the three DNAs (clean, extracted and refined) are used to create a DES model of the warehouse box moving experiment to test this proposition. After running the model with each of the three DNAs and comparing the results, it can be concluded strongly that in fact, the DES model built using the refined DNA resembles the real system more closely than the DES built with extracted DNA. This comparison is tested using three quantifiable parameters (i.e. time to inspect each box, variation in unit cost, and the inspector's idle time) and the results are illustrated in Figure II-15 through Figure II-17. The cost to calculate the variation in unit cost in Figure II-16 is calculated by considering the total labor cost (one worker and one inspector) according to the Bureau of Labor Statistics (BLS) (2015) data at $15.34/hour for worker and $33.92/hour for inspector.

The improvement in the output of the simulation model built from refined DNA as opposed to the one built using extracted DNA can be seen clearly in Figure II-15 through Figure II-17. For instance, per Figure II-15, the average discrepancy in inspection time (seconds) per box reduces from 23.8 between clean and extracted DNAs to only 7.4 between clean and refined DNAs. Similarly, as seen in Figure II-16, the discrepancy in unit cost is reduced from 52.8% (using extracted DNA) on average to 16.5% (using refined DNA). Both parameters show major improvement in the accuracy of the simulation output compared to ground truth values.

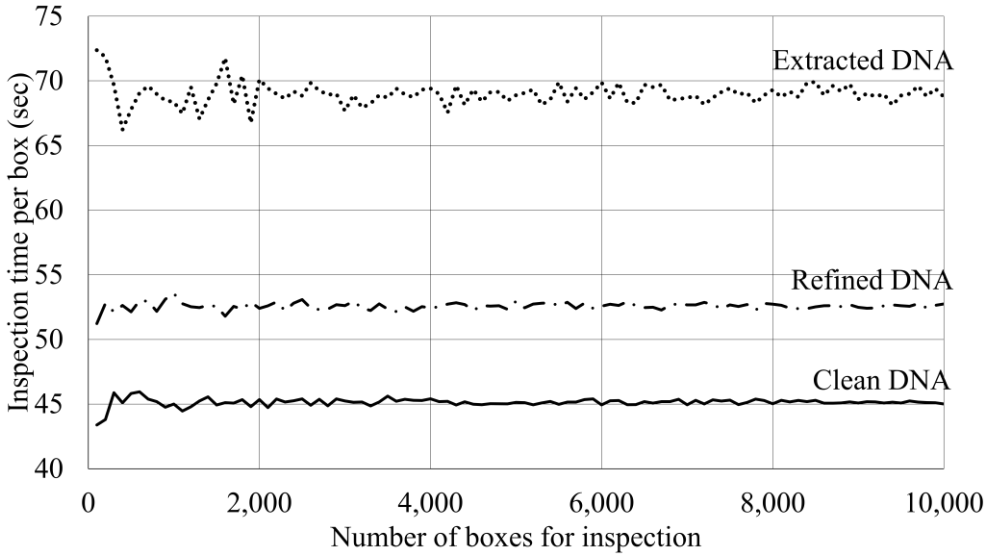


Figure II-15 Analysis of inspection time per box obtained from clean, extracted, and refined DNAs

Figure II-16 Analysis of unit cost discrepancy obtained from extracted and refined DNAs

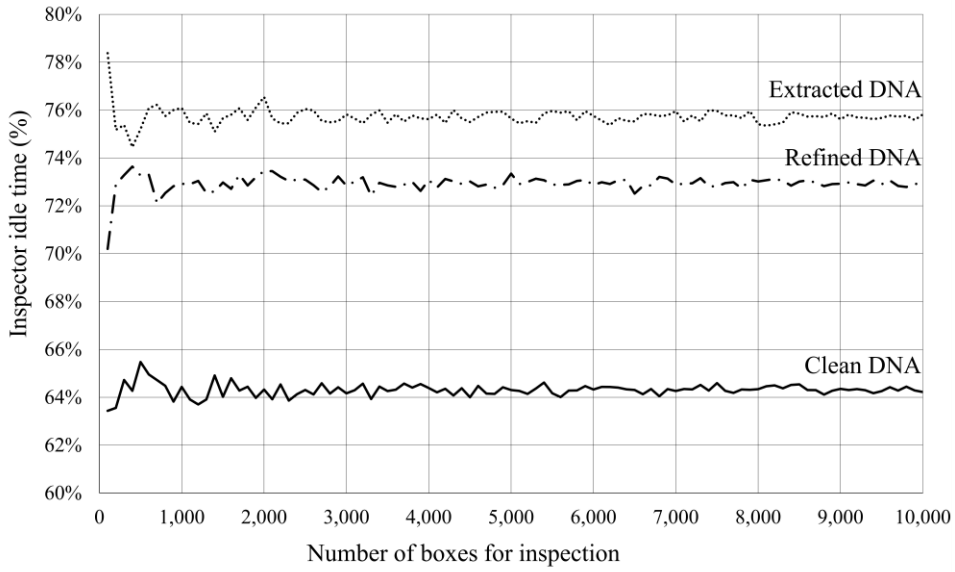


Figure II-17 Analysis of inspector's idle time obtained from clean, extracted, and refined DNAs

Finally, Figure II-17 shows a slight improvement in the inspector's idle time. In particular, while the simulation model built from the clean DNA shows that the inspector is idle 65.31% of the time, this value is calculated as 75.73% and 72.93% from the simulation models built from extracted and refined DNAs, accordingly. This is due to the fact that in the refined DNA the strength values of links associated with inspector activities did not significantly change during the optimization process compared to the extracted DNA, since most of the erroneous transitions took place while the inspector was idle and the worker was active.

II.6 Summary and conclusions

In construction projects, unforeseen site conditions, as well as the presence of other external factors such as such as adverse weather, change orders, lack of coordination, and resource misallocation often cause planned activity sequences and workflows to be altered. In order to incorporate these dynamic changes, DES modeling has evolved as a promising technique to formulate and study uncertainties in activity sequences and resource flows. However, DES tools often suffer from rigidity, user incompetence, and specificity of solutions, which prevent them to be widely adopted as reliable decision support systems. Moreover, most simulation systems cannot adapt to changes in project conditions as there is no systematic way to fully capture and incorporate heterogeneous process-level data into a simulation model.

In recent years, new opportunities to deal with this gap has been created with the advancement of sensing technology and increase in the amount of data available at project sites via the use of smart sensor grids. However, this new technology and the increased data is often unused despite great potential in project planning, implementation, monitoring, and control functions. The gap between data collection and data utilization remains significant and is further compounded by the noise inherent in sensor data. If used to create simulation inputs, this built-in noise can potentially propagate in the model and result in volatile outputs, further contributing to unreliable and inaccurate simulation results.

The work discussed in this Chapter aimed at investigating whether low quality sensor data captured by consumer-grade sensors can be still reliably used to generate stable simulation input models. In particular, sensor readings were processed first through a machine learning framework to detect activity sequences in a warehouse operation experiment, then the results were improved using an evolutionary algorithm. While activities (e.g. load, unload, lift, push, and pull) and their sequence in this experiment were relatively simple, it is worth noting that activities of this type are also the main building blocks of a large family of complex construction operations. For instance, a typical concrete placement operation involves activities that result in the formwork to be secured in place. In particular, formwork elements are first loaded on to a crate, pushed along a certain path, unloaded and then lifted to position. Similarly, in concrete placement, workers push and pull a concrete bucket, and screed the surface. Thus, activities used in the experiment presented in this paper belong to a representative subset of construction activities.

By coupling evolutionary methods (i.e. GA) and DES modeling, the uncertainty in activity precedence logic was refined, which in turn increased the fitness of activity transition matrix (a.k.a. DNA) from 0.76 to 0.96 (compared to the ground truth value of 0.97). thus, this validates that processing activity transition data through evolutionary algorithms can improve construction simulation models. The validity of this improvement was illustrated through the use of obtained refined data as inputs of a simulation model describing the operation. The output of this model was compared with the ground truth

using three metrics, namely total time to inspect each box, variation in unit cost, and inspector's idle time. Results showed that when the refined DNA was used as input, the simulation input was significantly improved than when the extracted DNA was used. In particular, improvements were seen in terms of time, cost, and productivity measures. Improvement in these and other parameters can significantly improve functions such as project scheduling and budgeting while providing clearer insight into the real system, potentially that can potentially improve the quality of workplace decisions impacting safety and health, ergonomics, resource allocation, and jobsite layout.

In the experiment presented in this paper, only one decision activity (i.e. inspection station) was used. In reality, however, construction operations may involve multiple decision-making points and more sophisticated activity transitions and resource interactions. For instance, a typical concrete operation involves several quality inspection stages (e.g. testing of concrete ingredients, rebar arrangement, formwork). Moreover, in this Chapter, it was presumed that the correct activity transition (benchmark), is the transition that was detected with the highest probability from the output of HAR. This was rooted in the basic assumption that sensor outputs are reliable to the most extent. However, in real world, there may be cases where a specific sensor or a subset of a larger sensor network are faultier than expected. As such, a better strategy must be established to identify the benchmark sequence from sensor readings

The main contribution of the work presented in this Chapter to the body of knowledge is a scientific methodology that facilitates the improvement of imperfect

(noisy) sensor data to cleaner datasets to increase stability of simulations that represent real engineering systems better. While a specific scenario was used to validate the developed methodology, the foundational mathematical and theoretical concepts can be adapted to other cases where sensor readings are needed to create input models for decision support systems and other modeling techniques in addition to DES. For instance, better data helps increase the R^2 value in regression analysis, improve the accuracy of extracted features in activity recognition, and reduce trajectory prediction error in path planning algorithms.

This framework has been shown to be effective in dealing with data on transition between different activities. However, its application is limited in the context of activity recognition in sequence of activities. Thus, the following Chapters deal with the improvement in activity recognition in the context of the larger sequence of activities, thus expanding the scope of application in natural phenomena in enabling greater use of simulations.

CHAPTER III

IMPROVING ACTIVITY DEPENDENCY DATA USING SEQUENCE ALIGNMENT

III.1 Introduction

Sequence alignment (SA) is a technique for evaluating the degree of similarity between two strings of data by using a series of heuristic or probabilistic methods to align one sequence with another (Rosenberg 2009). This approach was developed in the bioinformatics domain in the 1980s to enable the comparison of long deoxyribonucleic acid (DNA), ribonucleic acid (RNA) and protein sequences which could not be efficiently processed by using conventional algorithms. Given the complexity of biological data, the ability to determine the degree of similarity of a pair of biological sequences is of great importance in answering questions such as inferring the function or source organism of an unknown gene sequence, developing hypotheses about the relatedness of organisms, or grouping sequences from closely related organisms (Copasaro 2018). SA primarily relies on a series of applied mathematical algorithms (Sankoff and Kruskal 1983) for holistic sequential analyses that could provide insight into long sequences of protein and DNA. In a nutshell, the SA algorithm compares a target sequence (e.g. unknown data sequence) with a source sequence (e.g. known data sequence). The use of SA was expanded to other domains in the late 1990s (Abbott and Tsay 2000; Wilson 1998) primarily by social scientists (Abbott and Forrest 1986) to advance the analysis of socio-economic data by

producing normalized data trends and comparing each data point to the trend. Other applications of SA include the development of linguistics algorithms to generate sentence-level paraphrases from unannotated corpus data (Barzilay and Lee 2003), and tools for analyzing the sequential aspects within the temporal and spatial dimensions of human activities (Shoval and Isaacson 2007), all in an effort to transition from unitized analysis to contextual understanding that explores connections rather than attributes (Abbott 1995). Another application of SA was demonstrated in studying dynamic human interactions by Huang et al. (2010) who used passive radio-frequency-identification (RFID) data from objects (describing parameters such as location, motion, and orientation) to train a model to recognize various daily human activities in a home environment. The variations between different instances of the same person and different people performing same activities were dealt with by using flexible SA to recognize common patterns of change for each activity.

Along with the evolution of SA techniques, the need for dynamic programming platforms was recognized and fulfilled by solutions including Clustal (Higgins and Sharp 1988) which was later expanded to ClustalX for multi-dimensional alignment and ClustalG for social science data (Wilson et al. 1999). Each new solution not only did add and adapt features helpful to the application area of interest but also dealt with computational challenges both in processing time, and space and time required to run the alignment algorithms. For instance, while in 1988, aligning 4 sequences was deemed beyond the capability of the available hardware (Higgins and Sharp 1988), by 2004,

improvements in hardware and evolution of heuristic algorithms enabled aligning up to 1,000 sequences of an average length of 282 in only 21 seconds (Edgar 2004).

As discussed in the previous Chapter, despite the rapid advancement of sensing technology and vast availability of data in AEC/FM, data-driven decision-making is still in nascent stages mainly due to issues such as data quality, reliability, and timeliness, which are mainly rooted in the lack of processing framework, high upfront costs, and data loss and latency (Islam et al. 2012), noise and human errors (Zamalloa and Krishnamachari 2007), and the complex nature of many projects. Moreover, most applications tend to be inflexible to changes in ground conditions.

While the previous Chapter illustrated the potential of a new GA-simulation hybrid framework to improve the reliability of activity transition data, continuing along the path of adapting phenomena from nature to improve data quality, the research presented in this Chapter vies to explore and assess SA as an alternative approach to processing and recognizing patterns in collected data sequences, by deploying holistic measures of comparison between datasets instead of merely relying on attributes of individual data points.

III.2 Basics of the sequence alignment (SA) algorithm

Traditional quantitative measures such as data clustering that are used to compare sequences are based on Euclidian distance measurements (Abbott 1995). These measures use a point-by-point approach to analyze sequences, which can quickly turn into an

exponentially complex problem as each new data point is possibly a point of diversion where a new parallel problem with equal complexity is created. These methods also require the grouping of similar features that are often defined subjectively and do not evolve over the course of the analysis. In addition, any small shift in the elements of a sequence could produce different results. In contrast, SA deals with data sequences as a whole. As shown in Figure III-1, SA measures the degree of similarity between two sequences (a.k.a. “source” and “target” sequences), using three basic operations: deletion (where an element is removed from the target sequence), insertion (where an element is added into the target sequence), and substitution (where two elements are switched in the target sequence) (Shoval and Isaacson 2007).

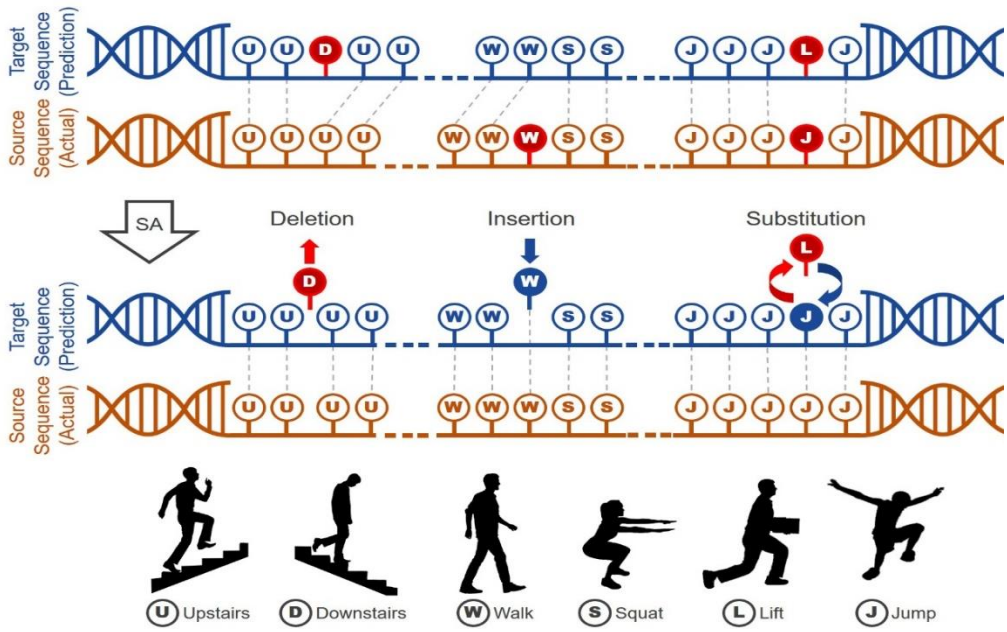


Figure III-1 The three primary operations in SA algorithm

In basic terms, the number of operations required to produce identical sequences is inversely related to the degree of similarity between the two sequences. In the example shown in Figure III-1, the two simple sequences are shown to vary in three elements. To produce identical sequences, these discrepancies are treated using the three operations of SA applied to the source sequence: deletion is used to remove the extra Activity ‘downstairs’, insertion is used to input a missing Activity ‘walk’, and the mismatch of Activities ‘lift’ (in the source sequence) and ‘jump’ (in the target sequence) is dealt with by substitution.

The platforms made to run SA are oriented primarily toward applications in bioinformatics and are thus mostly limited to 20 characters (corresponding to the number

of amino acids in human DNA) (Shoval and Isaacson 2007). Existing SA algorithms can be designed to converge globally (a.k.a. global alignment) or locally (a.k.a. local alignment). While the former aligns entire sequences, the latter considers regions of similarity in globally differing sequences and can thus be more resource intensive (Polyanovsky et al. 2011).

III.3 Research objective and contributions

While SA has been quite extensively used in other domains, its potential in AEC/FM in areas such as jobsite and facility management, building operations, energy performance, and fleet management remains limited. Operations within these dynamic systems can largely benefit from robust optimizations that target the detection and classification of complex, versatile, and spatiotemporal interactions between humans, equipment, and tools that together influence the overall efficiency of the process. Even with the proliferation of data, sensors, and reality capture tools, detecting such interactions with high fidelity for reconstruction in computer interpretable formats (e.g. simulations) is time consuming, inaccurate, and complex. Current methods such as RFID tracking, image and video recognition, manual inspection, barcodes tracking, and laser scanning require extensive initial investment for setting up and calibration, and call for advanced expertise for proper operation and maintenance (Kiziltas et al. 2008; Kopsida et al. 2015). Moreover, these methods may not support (near) real time processing (Park et al. 2013) which can negatively impact the timeliness and/or accuracy of resulting decisions. This

issue has been cited as a major obstacle to the widespread adoption of data-driven decision-making tools in construction (Becerik-Gerber et al. 2013). In light of this, the work presented in this Chapter seeks to create and test a new method that allows for a high-fidelity transformation of raw sensor data into contextual knowledge in order to test the hypothesis that the overall accuracy of activity recognition can be improved through sequence alignment. Such knowledge can be useful to describe the status/sequence of activities in a dynamic system, while also providing a basis for performance benchmarking and identifying areas of waste, mistakes, and inefficiencies.

III.4 Methodology

The experiment considered for illustrating the designed SA technique is a lab floor with multiple individuals labeled as $W1$, $W2$, $W3$, and $W4$ each wearing a smartphone on their dominant arm. Built-in smartphone sensors are used to collect time-motion data while subjects perform six activities, namely 'walk', 'lift', 'squat', 'walk upstairs' (or 'upstairs' in short), 'walk downstairs' (or 'downstairs' in short), and 'jump'. For each person, a complete cycle consisted of each of these activities performed in an arbitrary order. However, the first cycle was designated as the control cycle and the order of the activities performed was predetermined as 'walk', 'upstairs', 'downstairs', 'squat', 'jump', and 'lift' (which can be represented as a w-u-d-s-j-l sequence). Each person completed 4-6 cycles. The goal of this experiment is twofold: (1) collect and process time-motion data (acceleration, linear acceleration, and gyroscope) to identify activities

performed by each person using HAR, and (2) improve the accuracy of results by post-processing the output of HAR using the SA algorithm. A description of each step is provided in the following Sub-sections.

III.4.1 Human activity recognition (HAR)

The application of HAR techniques has been recently explored by some researchers in the construction domain (Golparvar-Fard et al. 2013; Yang et al. 2014; Akhavian and Behzadan 2016) to identify instances of major events (e.g. human activities) from sensor data using a host of machine learning (ML) algorithms. This identification takes place in training and testing phases (Dunham 2006; Harrington 2012) where an initial dataset is used to identify distinct features of the different classes (activities) and manually label them. Next, identified features are used to classify the testing dataset. A detailed account of the HAR step can be found in Nath (2017).

As related to the lab floor experiment described above, the dataset that contains one cycle of activities of a single subject (in this case, first cycle of *W*1) is considered as the training dataset while all other datasets (i.e. remaining 5 cycles of activities for *W*1, as well as all cycles of activities for *W*2, *W*3, and *W*4) are considered as testing datasets. This approach was chosen considering a key practical limitation in data collection; it may not be possible to collect enough training data from each and every participant, especially when the operation takes place in a large system with constantly changing spatiotemporal properties. Instead, it is more practical to collect training data from a small subset of

participants, train the classifier model using this sample dataset, and later apply the trained model to the entire group.

III.4.2 SA

The HAR algorithm is designed in a parallel study (Shrestha et al. 2018) to provide four elements as outputs: the source (ground truth) and target (recognized) activity sequences for the control cycle (i.e. first cycle of each person), the confusion matrix which is obtained by comparing the source and target sequences, and the activity sequence recognized for the remaining cycles for each person.

By nature, the confusion matrix is a proportional representation of different instances where an activity is either classified properly or miss-classified as another activity. This is illustrated by the two sample sequences shown in Figure III-2, and the resulting confusion matrix of Figure III-3(a). Rows in a confusion matrix represent ground truth activities whereas columns represent recognized activities. For instance, per Figure III-3(a), Activity A is identified correctly twice, Activity B is identified correctly twice, and Activity C is identified correctly 5 times. These instances are reflected in the diagonals of the corresponding confusion matrix of Figure III-3(a). The sum of non-diagonal elements in this matrix equals 9 which indicates that overall, activities are not identified correctly in 9 instances. For example, Activity C is misidentified as Activity A twice (in instances 6 and 12, as marked in Figure III-2). For simplicity, once the confusion matrix is built, values are expressed as percentages of the total instances of each activity. This representation is shown in Figure III-3(b)

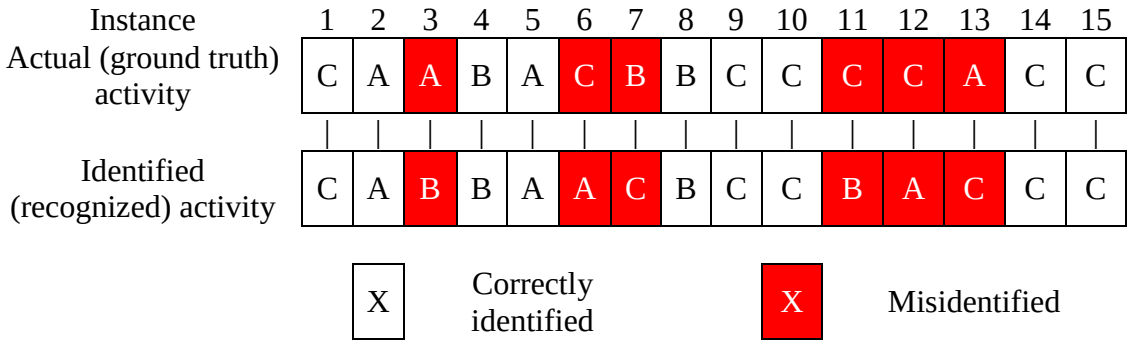


Figure III-2 Results of a hypothetical activity recognition scenario

	A	*B*	*C*
A	2	1	1
B	0	2	1
C	2	1	5

(a)

	A	*B*	*C*
A	50%	25%	25%
B	0%	67%	33%
C	25%	12%	63%

(b)

Figure III-3 Sample confusion matrix showing (a) absolute values, and (b) percentages

The output of HAR is very likely to contain errors due to various factors including inaccurate sensor readings, heterogeneous actions by workers, and classifier drift (e.g. due to under-fitting or over-fitting). This erroneous output comprises the input of the designed SA algorithm as shown in Figure III-4.

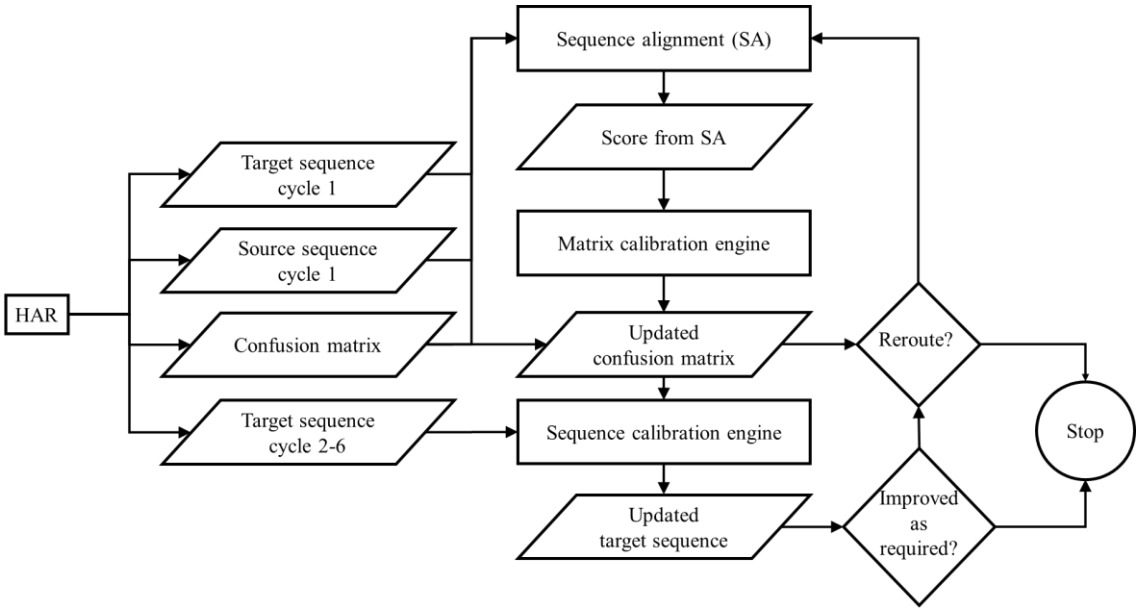


Figure III-4 Workflow for post-processing HAR results using SA

As Figure III-4 shows, in the initial SA algorithm, the source and target sequences are aligned in the 'sequence calibration engine'. This is done using a dynamic programming application known as the Smith-Waterman local alignment (Smith and Waterman 1981). This algorithm is based on the Needleman-Wunsch global alignment (Needleman and Wunsch 1970) focusses on comparing subsequences of all possible lengths and finding the optimal combination to maximize the similarity measure.. Each pair compared is classified as a match or a mis-match. In case of a match, a positive score is assigned to the pair whereas in the case of a mis-match, a negative score is assigned. The magnitudes of both scores are predetermined in the scoring matrix and can vary across the different pairs. These scores are cumulated across the different pairs and the highest scores in the matrix is determined to be the overall score of the alignment.

In the case of this application, in order to assign a numerical value to each match, the probability of the match from the confusion matrix is used as the basis for computation. In essence, the numerical score for a particular combination of activities is inversely proportional to the probability in the confusion matrix of the activity being identified correctly. Given the complex comparisons and possibilities of SA, the developed technique performs normalization of scores obtained from different alignments according to length. This is an important consideration in producing reliable results as different people may perform identical tasks at their own pace, thus resulting in sequences of unequal length that represent the same set of activities. The SA algorithm is run separately for each of the four instances of source and target sequences producing four individual scores. These scores are fed to the 'matrix calibration engine'. For each instance *i*, there is now a score (*i*) and a confusion matrix (*i*). These two datasets are used to create a cumulative confusion matrix (CCM) with element in the j^{th} row and k^{th} column calculated by Equation III-1.

$$CCM\ (j,k) = \frac{\sum_1^4 score\ (i) * confusion\_matrix(j,k,i)}{\sum_1^4 score\ (i)} \qquad \text{(III-1)}$$

In order to increase the accuracy of target sequences, the 'sequence calibration engine' identifies anomalous activities and replaces them with more probable substitutes. In this process, the percentage of instances in which other activities are misidentified as the anomalous activity is taken into account using the percentages in the confusion matrix.

These percentages are then used as weights to probabilistically pick the replacing activity. The higher the rate of confusion, the higher the chance of the corresponding activity being picked as the replacing activity. This cycle of alignment, calibration, and replacement is continued for several generations. In each iteration, the global fitness parameter (GFP) is calculated, as shown in Equation III-2.

$$GFP = \frac{\sum_{cycle\ 1\ to\ 6} correctly\ identified\ activity\ instance}{\sum_{cycle\ 1\ to\ 6} activity\ instances} \quad \text{(III-2)}$$

III.5 Results and analysis

The initial output of HAR classification is shown in the confusion matrix of Figure III-5. Here, Activities ‘idle’, ‘walk’, and ‘squat’ are classified with more than 90% accuracy, while Activity ‘downstairs’ is predicted with the least accuracy (45%) mainly because it involves physical movements like those of activities ‘upstairs’ and ‘walk’.

	Idle	Walk	Upstairs	Downstairs	Squat	Jump	Lift
Idle	97%	1%	1%	0%	1%	0%	0%
Walk	0%	91%	8%	1%	0%	0%	0%
Upstairs	0%	31%	61%	3%	0%	5%	0%
Downstairs	0%	10%	35%	45%	2%	8%	1%
Squat	2%	0%	5%	0%	92%	0%	0%
Jump	1%	1%	2%	5%	2%	86%	2%
Lift	1%	9%	2%	13%	0%	0%	76%

Figure III-5 Initial confusion matrix from HAR classification

Next, GFP was recalculated for 25 generations, as plotted in Figure III-6. While the probabilistic elements of the designed SA process produce some variability in the results, the value of GFP improves gradually. The best result, obtained in generation 24 achieved a GFP of 87.25% improving upon the initial value of 85.7% calculated directly from the confusion matrix produced by the HAR classifier model.

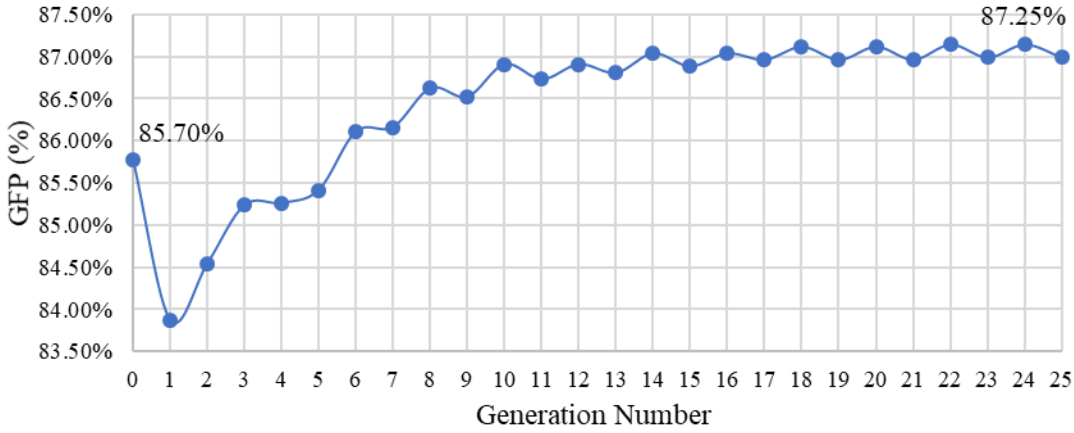


Figure III-6 Variation in global fitness parameter (GFP) over several generations

Figure III-7 shows several improvements in the confusion matrix. In essence, the rate of correct activity detection for ‘walk’ is increased by 4 % (from 91% to 95%), ‘upstairs’ is increased by 6% (from 61% to 67%), ‘downstairs’ is increased by 2% (from 45% to 47%), ‘jump’ is increased by 5% (from 86% to 91%), and ‘lift’ is increased by 1% (from 76% to 77%).

	Idle	Walk	Upstairs	Downstairs	Squat	Jump	Lift
Idle	94%	1%	1%	1%	1%	1%	1%
Walk	0%	95%	4%	1%	0%	0%	0%
Upstairs	0%	26%	67%	0%	0%	5%	2%
Downstairs	0%	5%	35%	47%	1%	8%	4%
Squat	0%	0%	3%	5%	91%	0%	1%
Jump	1%	2%	1%	4%	1%	91%	1%
Lift	0%	6%	1%	14%	0%	2%	77%

Figure III-7 Improved confusion matrix after SA implementation

III.6 Summary and conclusions

This Chapter presented SA, a bioinformatics technique, as an alternative post-processing approach to refining imperfections resulted from using raw sensor readings for HAR. The process started with data collection during which time-motion (acceleration, linear acceleration, and gyroscope) data were collected from built-in sensors of smartphones worn by several individuals who performed six common activities including ‘walk’, ‘lift’, ‘squat’, ‘walk upstairs’, ‘walk downstairs’, and ‘jump’. Raw data was then used as input of HAR to train and test classifier models. The output of HAR was consequently used as input to SA, to further refine resulting confusions in activity recognition, and improve the overall fitness of the HAR results. In general, the accuracy in predicting five of the seven activities was significantly improved (as shown by the diagonal elements of confusion matrices in Figure III-5 and Figure III-7). In addition, the GFP (overall measure of fitness) of HAR results increased after the application of SA.

The work presented in this Chapter expands the scope of application of phenomena in nature to improve the utility of sensor data as input for construction simulation. Improved accuracy of activity recognition increases the stability and reliability of simulation models and thus their use in the decision-making process. However, in its current application the algorithms is limited to improving the output obtained from HAR algorithm, thus maintaining the need for a two-step process. In the following Chapter an exploration of an expanded version of SA is presented with the goal of integrating the HAR and SA process in a multi-dimensional sequence alignment (MSA) process that performs both functions.

CHAPTER IV

REFINING SENSOR LEVEL DATA USING MULTI-DIMENSIONAL SEQUENCE ALIGNMENT

IV.1 Introduction

Multi-dimensional sequence alignment (MSA) is an expanded form of sequence alignment (SA) in which the various attributes (i.e. dimensions) of source and target sequences are compared separately and an aggregation of the scores calculated from comparison in each of the attributes is used in the classification process. Similar to previous Chapters, this implementation is also inspired by phenomena found in nature and aims at streamlining the previously developed two-step human activity recognition (HAR) coupled with SA (a.k.a. HAR-SA) discussed in Chapter III, by proposing a single-step processing workflow to process raw body-mounted sensor data. With the expansion of computing capabilities and scope of SA beyond bioinformatics, researches have recently sought new methods to adapt the powerful concepts of sequential data comparison into novel applications. In this regard, MSA was proposed as a way to apply SA to datasets with more than one relevant attribute. Moreover, while applying individual SA to each attribute gives researchers useful yet partial information, combining and relating the scores obtained from SA in a multi-dimensional framework provides a more extensive picture (Joh et al. 2002).

With this in mind, the objective of the work presented in this Chapter is to investigate whether the process of generating simulation input models from raw sensor data can be further improved through a single-step MSA implementation without compromising the reliability of the generated simulation input models. This proposition is evaluated in the context of the results obtained from simulation models built from the two approaches (HAR-SA and MSA) and comparing them to the simulation results obtained using the ground truth information as input.

IV.1.1 Comparing the classification principles of MSA and HAR

The process of classifying activities using MSA shares several of the same steps successfully implemented in supervised (inductive) machine learning (ML); in principle, both algorithms use previously labeled data to classify unlabeled data. In Chapters II and III, HAR was successfully implemented using the principles of supervised learning (Nath 2017). However, the main point of divergence from HAR in MSA implementation lies in how prior knowledge is utilized in the identification. In particular, HAR is based on the discovery and selection of a variety of features that can differentiate between various classes in any given feature window. It posits that this feature space, defined in the context of a window, can be representative of the various classes and thus differentiation is possible. On the other hand, MSA is based on the principle that the information in data in a sequence of windows is representative of the various classes. Thus, MSA posits that the relationship between the different elements of the sequence being classified can be a basis of categorization. The similarities and differences between the two implementations can

be better described using the hypothetical comparison in Figure IV-1. In this Figure, a target sequence is compared against a source sequence using both HAR and MSA. For HAR, three features are shown to have been extracted from the source sequence and the target sequence, whereas for MSA the continuous sensor amplitude data from both sequences is first discretized and then compared. In this process, both algorithms involve the extraction of information from raw sensor data. However, while in HAR the identified features have a constant value throughout the window, in MSA the variation in the sequence within the window is harnessed. Thus, while HAR requires the computation of a large number of features to correctly identify class labels, MSA does not require the extensive feature extraction as comparisons are implemented directly on the discretized sensor data. One major advantage of this difference in approach is that while the size of the windows, the starting point of each window and the degree of overlap among windows is pre-determined in the HAR implementation, the starting point in time ($t = 0$) for each comparison in MSA can be easily altered. In the case of HAR, changing the starting point or the overlaps would require a re-extraction of features with different window starting points and lengths.

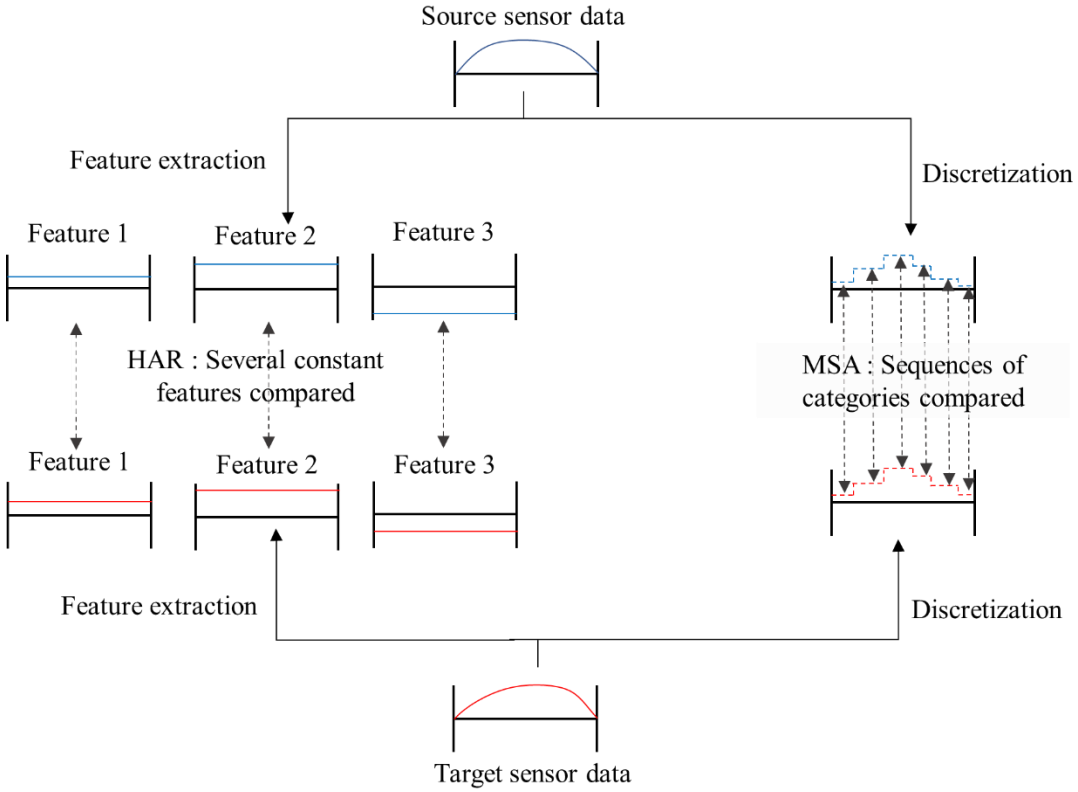


Figure IV-1 Illustration of the similarities and differences between HAR and MSA

Furthermore, as both HAR and MSA are supervised ML algorithms, they require the conversion of continuous sensor data to a discrete feature space (Dougherty et al. 1995).In the case of HAR this is achieved by applying statistical operations to data points within a particular window, whereas in the case of MSA, frequency-based binning is used in the context of the entire dataset, as better explained in the following Sections.

IV.2 Research objectives and contribution

As highlighted above, MSA is a promising framework for processing multi-dimensional data with a number of attributes. However, the applications of MSA is yet to be explored as a potential framework of processing data and supporting data-driven decision-making in the construction domain, as its existing applications have been very limited in scope. Thus, this Chapter seeks to develop a novel utilization of the MSA technique where different streams of raw sensor data are processed directly to identify the activities performed. This application not only does expand the general scope of MSA, it also simplifies the overall framework by eliminating the need for pre-processing in the algorithm relied upon in previous Chapters. Furthermore, the contribution is not limited to the pre-processing stage, since the proposed framework also enables the post-processing comparison (following HAR) of activity sequences using more than one (as was the case in simple SA) attributes of data.

IV.3 Methodology

In general, MSA is implemented in order to tag an activity label to an unknown sequence of raw sensor data. This is achieved by aligning the new (unknown) data sequence with several data sequences each representing a known activity, as schematically represented in Figure IV-2. Both the data stream representing the unknown activity (a.k.a. target sequence) and the data streams of known activities (a.k.a. source sequences) are assumed to have k dimensions. In the context of sensor data, a dimension refers to a

specific sensor reading along a certain axis (e.g. accelerometer readings along the *x* axis; gyroscope readings along the *y* axis). Using this convention, since an accelerometer collects data along *x*, *y*, and *z* orientations, its data is said to be three-dimensional. If built-in smartphone sensors are used to collect time-motion data, the number of dimensions in collected data can be calculated by multiplying the number of sensors by the number of axes along which data are collected. For example, if a smartphone's accelerometer, magnetometer, and gyroscope (3 sensors) are used to collected data in *x*, *y*, and *z* directions (3 axes), then the data is said to have 9 dimensions. Clearly, if more than one data collection unit is used, the number of dimensions will increase accordingly. For instance, Barshan and Yuksek (2014) used five sensor units mounted on the torso, left arm, right arm, right leg, and left leg, and collected data from accelerometer, gyroscope, and magnetometers, thus resulting in a dataset with 45 dimensions (5 multiplied by 3 multiplied by 3).

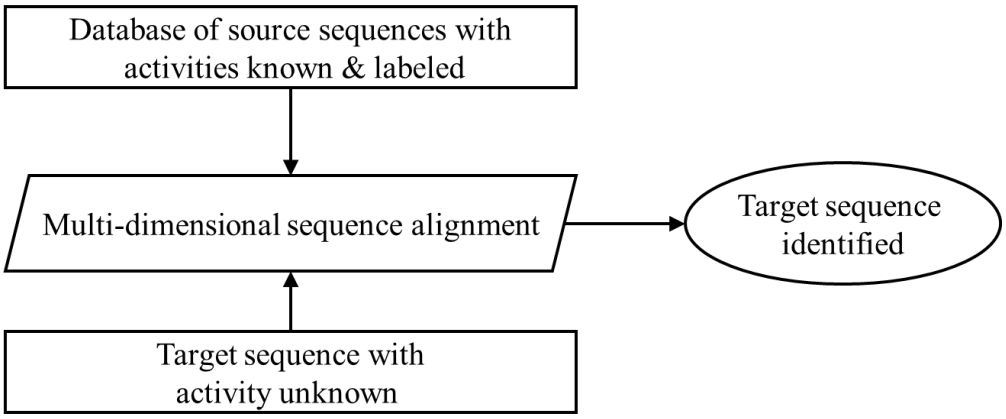


Figure IV-2 General schematic representation of MSA workflow

During the alignment process, data from each dimension of the target sequence is compared against the data from the respective dimension in all of the source sequences. A sequence alignment score is obtained from each comparison, and all scores are collectively used to evaluate the activity of the target sequences. For instance, if k-dimensional data is used, there are l seconds of sources sequence data available for each of the x possible activities and each comparison window is s seconds long, in total $(k \times l \times x)/s$ comparisons are made to identify the target sequence window. The details of how the scores are calculated and assessed are elaborated later in the following Sub-sections.

Figure IV-3 provides an overview of the steps involved in the classification of the unknown target sequences. In the discussion that follows the methodology is divided into two main phases: the training phase where the parameters of comparison are identified using the known sequences, and the testing and classification phase where the unknown sequences are classified.

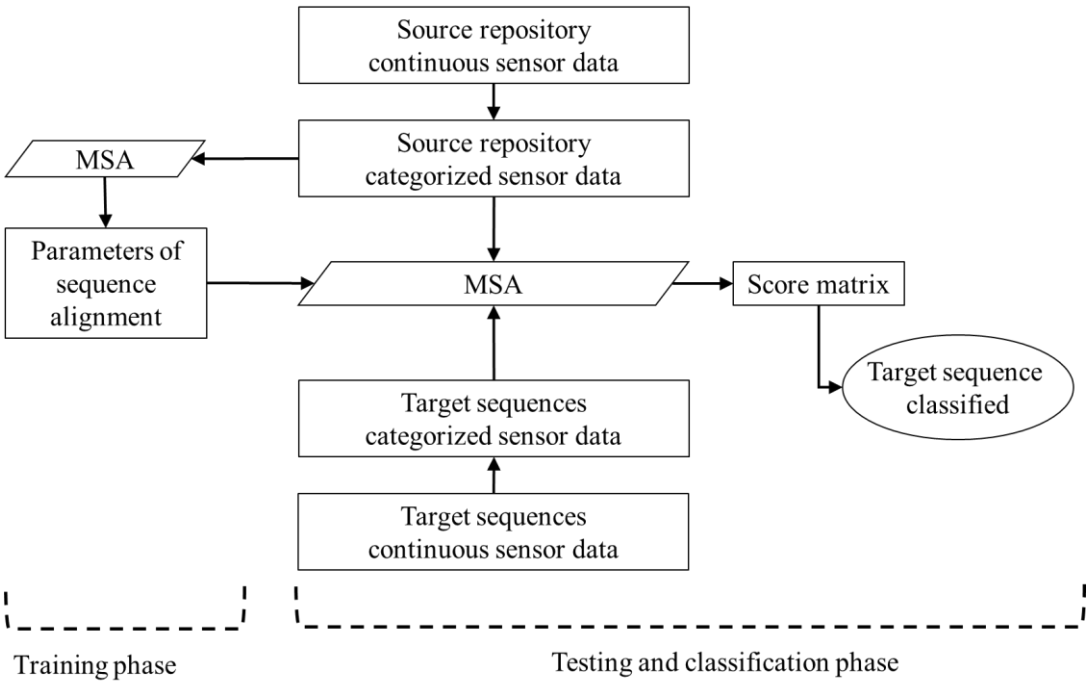


Figure IV-3 Detailed illustration of the phases of MSA workflow

IV.3.1 Training phase

IV.3.1.1 Building a repository of source sequences

The first step in the testing phase of MSA implementation is to build a repository of reference sensor data information for use in deriving the source sequences. This is done by collecting multi-sensor time-motion data from experiments in controlled settings where subject(s) perform a series of activities. Since the ground truth is known, collected data can be labeled accurately and used later as benchmark information.

IV.3.1.2 Normalization of raw continuous data collected from sensors

The information collected from the sample experiments consist of continuous raw sensor data which expresses the amplitude recorded by sensors in each of the dimensions. This presents two problems. First, the application of SA is based on the principle that the overall nature of activities (i.e. the ergonomic motions during the activities and thus the trends in the slope of the data collected by various sensors) remains the same even when performed by different people under different circumstances. However, amplitude data collected from sensors are susceptible to significant variation due to sensitivity to the intensity and manner in which different people perform even similar activities. The second problem is borne out of the fact the SA is primarily an ordinal comparison algorithm, making it unsuitable in its original form to be applied to the continuous sensor data.

In order to address these two issues, the raw sensor data is put through a series of steps intended to normalize the variations in amplitude and discretize the sensor data so that it can be used in SA. First, the general trend in the data is incorporated by calculating the slope (of amplitude over time) between successive data points (i and i - 1) using the formula shown in Equation IV-1 This formula is applied to all data points across all dimensions and activities.

$$Slope\ (i) = \frac{Amplitude\ (i) - Amplitude\ (i-1)}{t(i) - t(i-1)} \qquad \text{(IV-1)}$$

IV.3.1.3 Discretization of the continuous sensor data

The second problem mentioned above is dealt with by building a representative dataset of categorical data. This is achieved by a process known as discretization which enables the quantization of continuous attributes. (Liu et al. 2002). Discretization is used extensively in ML in order to reduce data while improving the prediction accuracy of the algorithms, especially in inductive (supervised) learning applications.

Liu et. al. (2002) provides an overview of different discretization methods used in data science for different purposes. In this research, for implementing MSA at raw sensor level data and given the structure of the available datasets, a global discretization in a direct equal-frequency splitting framework is implemented. Global discretization incorporates all the information (e.g. equal-interval-width discretization, equal-frequency-per-interval discretization, minimal-class-entropy discretization) available in the entire space, thus resulting classification can be reliably evaluated in the context of the entire dataset (Chmielewski and Grzymala-Busse 1996). Moreover, frequency splitting is appropriate in negating the skewing of weighted measures that outliers can cause. Frequency splitting uses measures that rely on positional information such as percentiles to determine cut-off points between categories. Furthermore, the number of discrete categories is predetermined as 20 in the algorithm designed and implemented in this research. Therefore, a direct method of discretization in which the number of categories is specified is used for implementing MSA in this Chapter. Overall this framework can be classified as binning; it essentially discretizes the data into 20 separate bins by examining

the rank of each data point in the context of all available sample data. This implementation of univariate discretisation (i.e. discretization using the information available within one attribute of data) has been validated as a method of data pre-processing in ML by a robust body of work that has evaluated the final results obtained using the discretized input, even in noisy environments (Han and Kamber 2011; Kotsiantis et al. 2006; Liu et al. 2002; Pfahringer 1995).

Using the foundation laid above, percentile ranks are used as cut-off points in the discretization of the data points. For each data point in a particular dimension, the percentile rank is assigned by comparing that data point against all available data points in that particular dimension. This process minimizes the role the outliers play in defining the overall nature of the distribution while bridging the gap between continuous data and discrete data. Finally, each data point in the dataset is classified as one of 20 possible categories based on the percentile rank, which each category consisting of 5 percentile ranks. For instance, the first category includes data in the 0^{th} to the 5^{th} percentiles, the second category includes data in the 6^{th} to 10^{th} percentiles, and so on. The number of categories is limited to 20 since existing SA algorithms were originally designed for bioinformatics applications to compare sequences of amino acids. Since most organic matter is made up 20 basic amino acids (Simoni et al. 2002), current SA applications are limited to an alphabet representing the 20 amino acids. Consequently, in this research, all data points are classified into one of the 20 ordinal categories each represented by its own symbol.

IV.3.1.4 Identification of optimal parameters of SA through cross validation

Various parameters affect the scores obtained from each SA comparison and this in turn affects the accuracy of classification of the activities. Thus, the selection of the optimal combination of parameters is important in achieving the required accuracy in classification in order to produce reliable simulation input models. Robust results in the testing phase can be achieved by choosing a set of parameters that perform best across different datasets, thus, a cross validation is run among the source repository. The dimension of cross validation (i.e. the number of folds), is a function of the type of classification (subject-dependent vs. subject-independent) and the structure of the dataset. Details are discussed in later Sections.

In each iteration of the process, a portion of the data is designated as test sequences and compared against the rest of the data. Within each iteration, the SA algorithm is run with a series of combinations of different parameters and the accuracy of the activity classification is recorded. At the conclusion of the cross validation, the combination of parameters which produced the highest average accuracy is selected. The parameters that can be varied are enumerated below.

- The time interval of a single window which affects the number of data points in a sequence can be altered. For instance, in a 25 Hz dataset, a window spanning 1 second will have 25 data points whereas a window of 3 seconds will have 75 data points. In order to allow for the different cycle times of each activity, the time windows within each activity can be altered separately as well.

- The values in the scoring matrix (i.e. scores) can also affect overall accuracy. A score generally increases for a positive match and decreases in case of a negative match. Here, a number of different combinations in the ratios of positive and negative matches can be tested. Moreover, since the comparison is based on categories derived from continuous data, the distance between the various categories also has significant implications in terms of the general trend. For instance, a mismatch between categories representing the 4^{th} percentile and 6^{th} percentile would have minimal significance when compared to a mismatch between categories representing the 5^{th} percentile and 97^{th} percentile. Thus, the negative score assigned to the former mismatch would presumably be different than the one assigned to the latter mismatch. A variety of techniques can be used to examine the best relationship between the magnitude of negative scores for the various mismatches and the magnitude of positive scores for the various matches.
- The number of source sequences against which a target sequence is compared is also examined. While intuitively having more information by comparing the target sequence with all of the available source sequences might be deemed better, this can also be tested mathematically as in theory, comparing against less than the maximum number of available sequences could as well produce better results.
- In a dataset collected from an uncontrolled environment, it is difficult to computationally identify when a new activity cycle begins or even how long a typical cycle of an activity is. Moreover, this can vary across instances and among different

people. In order to minimize the effect of out-of-phase SA comparisons, overlaps between different comparison windows can be used to reduce the number of such comparisons. This process is also essential to identify the most useful metric of overlap.

IV.3.2 Testing and classification phase

The MSA phase of the workflow shown in Figure IV-3, is performed in order to identify the alignment scores between the target sequence of an unknown activity and the source sequences of known activities.

Before this phase is conducted, the target sequence is normalized using slope values calculated by Equation IV-1, and subsequently categorized into 20 ordinal categories using the same methodology used for the source sequences, as discussed in Sub-section IV.3.1.3. The only difference in the implementation of the process arises when determining the percentile ranks of the target sequence slope values. Since the algorithm is designed with the assumption that the target sequence is classified continuously (i.e. as new data on the target sequences comes in, it is classified in near real time). This implies that at the time of the target window classification, the data available for target sequence window can be limited to a only few windows. Thus, due to this limited size, the available target sequence dataset cannot be relied upon to provide an accurate assessment of the percentile ranks of the unseen data points. Thus, the percentile ranks of the target sequence data points are identified in the context of the available source sequence values.

Each target sequence window of a specified time interval (determined using cross validation within the source repository) is compared against several source sequences windows in each dimension using simple SA, with each comparison producing an alignment score. The number of scores obtained is a function of several parameters discussed in the previous Sub-section. Assuming that for each activity, *x*, *l* seconds of data is available in each dimension, *k*, and that the ideal window size is determined to be *s* seconds ($s < l$), the total number of available source sequence windows (N_{ss}) is given by Equation IV-2.

$$N_{ss} = \frac{\sum_x l \times k}{s} \tag{IV-2}$$

In this case, each SA comparison is conducted with sequences of $s \times f$ data points (*f*: data collection frequency), and corresponding scores ($S_{x,n,k}$) are used to generate a three-dimensional matrix where each cell represents a particular activity, *x*, a particular window, *n* in the source sequence, and a particular dimension, *k*. Next, for each pair of *x* and *n*, the dimension sum score (SS^k) is calculated by summing individual scores across all dimensions. This is shown in Equation IV-3

$$SS_{x,n}^{k} = \sum_k S_{x,n,k} \tag{IV-3}$$

This process has also been illustrated Figure IV-4 which shows a particular target sequence window being compared against a particular source sequence window. Each of these comparisons is repeated for each of the activities and dimensions to obtain a score matrix for that particular target sequence window and source sequence window across all

the activities and dimensions. Figure IV-4 illustrates a scenario where there are five possible activities and the data has five dimensions.

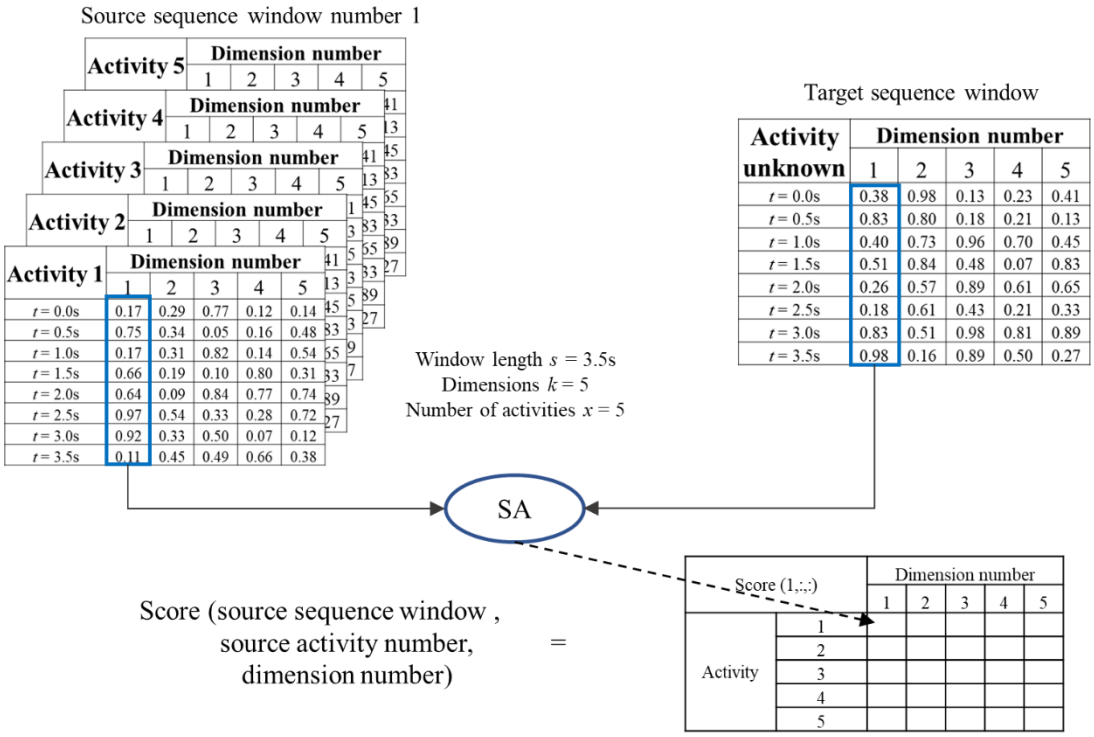


Figure IV-4 Illustration of formulation of the score matrix containing SA scores for a particular target and a particular source window across activities and dimensions

Continuing from Figure IV-4, the sum of the scores across the different dimensions yields SS^k, as illustrated in Figure IV-5. The calculation of SS^k is done for every labeled window of each activity (in each of the source sequences). Using this holistic picture of the relationship between the source sequence window and the target sequence window, the comparison yielding the highest SS^k determines the activity label of the unknown

window in the target sequence. Since, there are N_{ss} source sequence windows for each activity in the comparison, this step identifies N_{ss} activities, one for each of the source sequence windows. Finally, windows (out of a total of N_{ss}) bearing the same label are counted, and the activity (out of all possible candidates) with the highest count is selected as the label for the target sequence window. Alternatively, the counts can also be used in expressing the probabilities of the target sequence window being of a particular activity. In this regard, the count of windows with similar label divided by N_{ss}, expresses the probability of the target sequence window being of that label. For example, if all N_{ss} source sequence windows are identified as Activity 2, then according to the scores, the probability of the target sequence window being Activity 2 is 100%. However, if only 75% of windows are identified as Activity 2 and the remaining 25% are identified as Activity 5, then it can be stated that there is a 75% chance that the target sequence window is Activity 2, whereas there is a 25% chance the target sequence window is Activity 5.

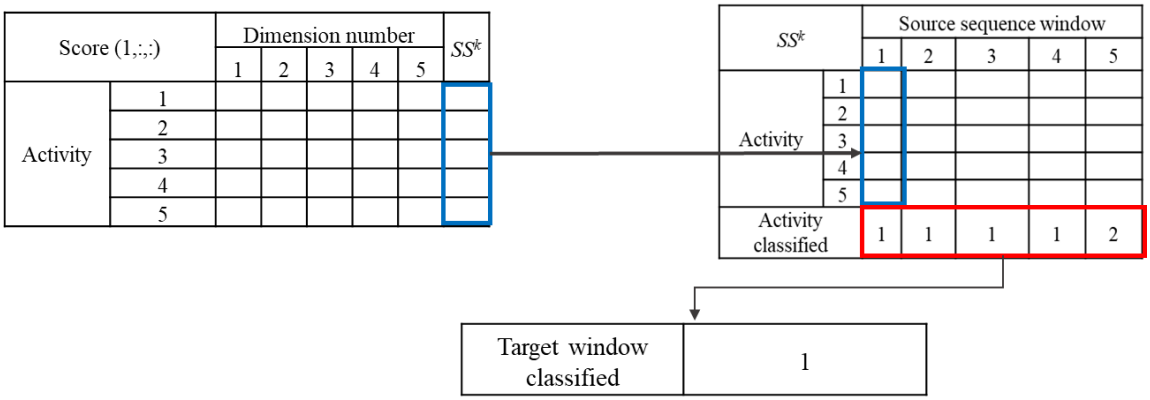


Figure IV-5 Classification of the target sequence window through calculation of SSk and classification with respect to each source sequence window

IV.4 Results and analysis

IV.4.1 Description of the input dataset used

The methodology developed in the previous Section is tested using a publicly available dataset (Barshan and Yüksek 2014) available from the University of California Irvine (UCI) Machine Learning Repository (Lichman 2013), containing sensor recordings collected for general applications in HAR. The dataset comprised of data from 8 subjects (4 male and 4 female, between the ages of 20 to 30) who performed five daily activities, namely ‘standing’, ‘walking upstairs’, ‘walking’, ‘running on a treadmill’, and ‘jumping’ in indoor (a sports hall, building) and outdoor areas at Bilkent University, Turkey.

Each activity was performed for 5 minutes by each subject and the data was collected at 25 Hz using 5 sensor units for each person mounted on the right arm, left arm, left leg, right leg, and torso. Each sensor unit collected accelerometer, gyroscope, and

magnetometer data along *x*, *y*, and *z* orientations. The main features of the dataset are summarized in Table IV-1.

Table IV-1 Summary of the different parameters of the input dataset

Category	Count	Description
Activities	5	standing, walking upstairs, walking, running on a treadmill, jumping
Subjects	8	4 males, 4 females, 20-30 years old
Data units	5	Mounted on right arm, left arm, right leg, left leg, and torso
Sensor types	3	accelerometer, gyroscope, and magnetometer
Orientation	3	along x, y, and z axes
Dimensions	45	5 data units × 3 sensor types × 3 orientations = 45
Frequency	25 Hz	
Data duration	5 min.	Per activity per person
Total data points	300,000	8 people × 300 sec. × 25 Hz × 5 activities = 300,000

IV.4.2 Evaluating the effectiveness of MSA for activity identification

The UCI dataset is used to evaluate the effectiveness of MSA to identify activities under two conditions: subject-dependent classification, where the training sample and testing sample are collected from the same subject, and subject-independent classification in which the training sample and testing sample are collected from different subjects. The algorithm was implemented Texas A&M University High Performance Research Computing (HPRC) clusters. In particular, the Ada cluster which comprises of an Intel

x86-64 Linux cluster with 852 compute nodes with each node containing an Intel Xeon 2.5GHz E5-2670 v2 10-core processor was utilized (HPRC 2018). In this implementation, up to 20 cores are used simultaneously.

IV.4.2.1 Subject-dependent classification

The data available for each subject is first divided into a training sample and a testing sample. In particular, 60% of the available data is designated as training sample, whereas the remaining 40% is used as testing sample. Considering the attributes of the UCI dataset as listed in Table IV-1, this translates into 15 minutes of training sample, and 10 minutes of testing sample for each subject.

The training sample is then used to identify the proper parameters for testing. At this stage, a 5-fold cross validation is implemented to identify the optimal combination of scoring matrix and the window length using data from each possible combination of sensors. Next, the identified parameter combination is used to classify the testing sample for each of the subjects. The confusion matrix incorporating the results obtained for all subjects in a 45-dimensional MSA using data from all 5 sensors is presented in Figure IV-6. As this Figure shows, Activities ‘standing’, ‘walking upstairs’, ‘walking’ and ‘running’, are classified with very high accuracy (100%, 99%, 100%, and 100%, respectively) whereas the accuracy with which Activity ‘jumping’ is classified is slightly lower, since this activity was confused with Activity ‘walking upstairs’ in ~6% of instances.

	Standing	Walking Upstairs	Walking	Running	Jumping
Standing	100%	0%	0%	0%	0%
Walking Upstairs	0%	100%	0%	0%	0%
Walking	0%	1%	99%	0%	0%
Running	0%	0%	0%	100%	0%
Jumping	0%	6%	0%	0%	94%

Figure IV-6 Confusion matrix obtained for subject-dependent MSA activity recognition using data from all 5 sensors in a 45-dimensional SA

The variation in the computation time required to implement the algorithm with the number of dimensions used in SA is also investigated in conjunction with the variation in the accuracy of the classification. In essence, data from various available sensors is used in different combinations to examine a total of 31 unique combinations, as tabulated in Table IV-2. Among these 31 combinations, 5 include data from 1 sensor, 10 include data from 2 of the 5 sensors, another 10 include data from 3 of the 5 sensors, 5 include data from 4 of the 5 sensors, and finally the last combination include data from all 5 sensors.

Table IV-2 Number of combinations using data from a given number of sensors and the dimension of SA performed

Number of sensors used	Number of combinations	Dimensions of SA
1	5	9
2	10	18
3	10	27
4	5	36
5	1	45

The results of this examination of the variation in the combination of sensors and the accuracy of classification obtained from classifying the target sequences is presented in Figure IV-7. In this Figure, the dashed line represents the average accuracies obtained for a particular sensor combination across the subject dependent classification of 8 subjects. Moreover, the bold line represents the average accuracy of classification obtained across the different sensor combinations that use the same dimensional SA. For instance, sensor combinations 1 through 5 used data from one sensor or 9-dimensional data each, hence, the value of the bold line (i.e. 95.5%) at combinations 1 through 5 represents the accuracy of classification using 1 sensor across all 8 subjects. It can be observed that changing the sensor combinations induces variation in the accuracy of classification, even when using data from the same number of sensors. For example, while sensor combinations 1 and 2 used data from one sensor each, the accuracy of classification is more than 98% for the former and less than 91% for the latter. Further, it is observed that the volatility decreases when the average accuracy of classification across the different dimensional SA is considered as a measure of performance. For instance, Figure IV-7 shows that while the accuracy of classification obtained from data of all 5 sensors is 98.3%, this value decreases by only 2.4% when using data from only one sensor.

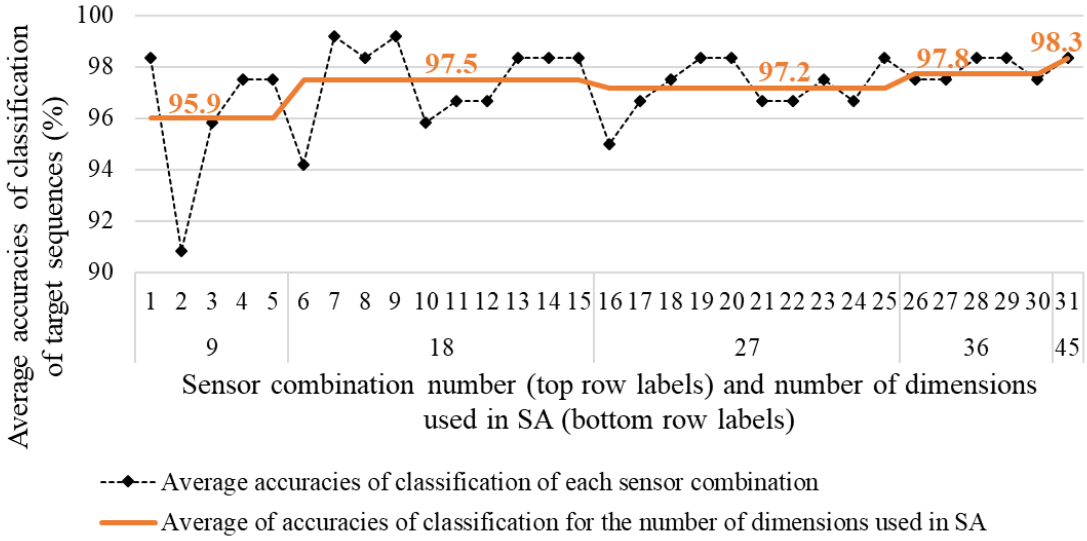


Figure IV-7 Accuracy of classification of target sequence activities with different training samples and SA data dimensions for subject-dependent classification

Next, the variation of accuracy of classification among different subjects is examined to obtain granular observations. For each of the subjects, the combinations enumerated in Table IV-2 are used and the obtained average accuracy of classification for each number of sensors is displayed in Figure IV-8. In this Figure, dashed lines represent the accuracy of classification for each subject while the solid line represents the average accuracy of classification for all subjects. Results indicate that in most cases, the accuracy of classification is quite high, and a high average accuracy is maintained. For instance, the average accuracy of classification across different subjects remains above 95% for each of the number of sensors, and in 212 out of the 248 total combinations examined (31 combinations for each of the 8 subjects), the accuracy of classification is more than 90%. However, for a small number of subject-sensor data combinations, accuracy is low. For

instance, the accuracy of classification for subject 2 is 73% when using data from sensor 2 as opposed to 91% overall for all subjects using data from the same sensor.

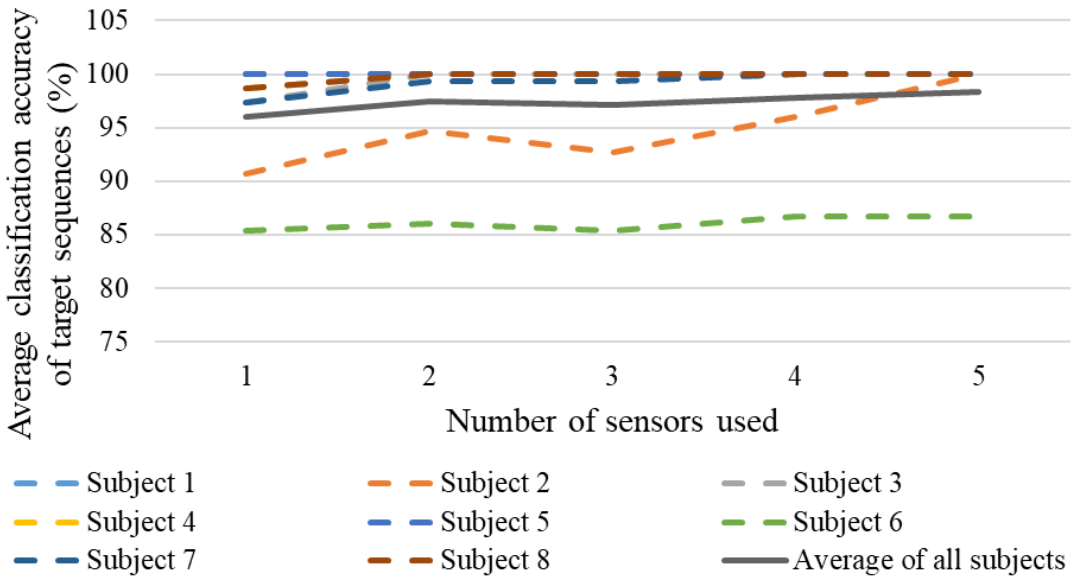


Figure IV-8 Accuracy of classification of target sequence activities with different subjects and SA data from different number of sensors for subject-dependent classification

The findings presented in Figure IV-9 also reveal that the computation time increases linearly in relation to the number of dimensions of comparison, however, the variation in accuracy (vertical bars in Figure IV-9) is less uniform. In this Figure, the right vertical axis shows time taken by the algorithm to train 120 minutes of data and subsequently classify 80 minutes of data using the identified parameters. According to results, while an increase in the number of dimensions generally improves the

classification accuracy, the improvement is minimal as the accuracy obtained when using only 9 dimensions is already quite high at 95.9%. Using all possible 45 dimensions, this accuracy increases only by 2.4% to a new value of 98.3%.

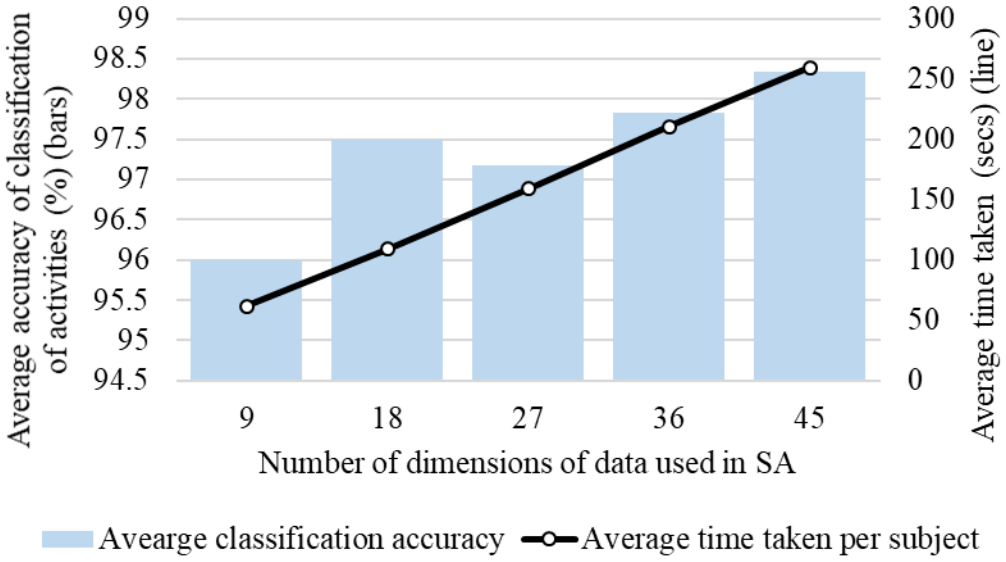


Figure IV-9 Accuracy of classification of target sequence activities and the average computation time required with different SA data dimensions for subject-dependent classification

IV.4.2.2 Subject-independent classification

The validity of the designed methodology is also tested for subject-independent activity classification using a similar breakdown of training and testing data as used in the previous Sub-section. In particular, data from 4 of the 8 subjects are designated as the training sample, whereas the data from the remaining 4 subjects are designated as the test

sample. The selection of the four test and training subjects is done randomly and different combinations of 4 people in the training sample are evaluated. In total, this process is conducted for 8 different combination of training and test samples. The confusion matrix incorporating the results obtained for all subjects in a 45-dimensional SA using data collected from all 5 sensors, in all of the combinations of testing and training subject groups is presented in Figure IV-10.

	Standing	Walking Upstairs	Walking	Running	Jumping
Standing	100%	0%	0%	0%	0%
Walking Upstairs	0%	100%	0%	0%	0%
Walking	0%	8%	92%	0%	0%
Running	0%	0%	0%	100%	0%
Jumping	0%	2%	9%	0%	89%

Figure IV-10 Confusion matrix obtained for subject-independent MSA activity recognition using data from all 5 sensors in a 45-dimensional SA

Comparisons between the results obtained from subject-independent classification (Figure IV-10) and subject-dependent classification (Figure IV-6) indicate that for most activities, the classification accuracies are comparable, whereas the accuracy decreases in the case of subject-independent classification for some of the activities. For example, Activities ‘standing’, ‘walking upstairs’, and ‘running’ are classified with extremely high accuracy in both scenarios, however the accuracy of classification of Activity ‘walking’ and ‘jumping’ decreases by 7% (from 99% to 92%) and 5% (from 94% to 89%),

respectively in subject-independent classification. This can be attributed to the variation in how different subjects performed the same activities. This decrease also reaffirms that variations in how subjects perform activities can affect the overall accuracy of activity classification.

Next, in order to examine the effect of the variation among different subjects, the algorithm is run several times with different combinations of training and test samples, using data collected from different combinations of sensors, as shown in Table IV-2. The accuracy of classification using data from each sensor combination is presented in Figure IV-11. In this Figure, the dashed line represents the average accuracy of classification across the 8 combinations of training and testing sequences, and the solid line represents the average accuracy of classification for each number of dimensions. For instance, only 1 sensor is used in combinations 1 through 5, which results in a 9-dimensional SA. For these combinations, the average classification accuracy is 88.2%.

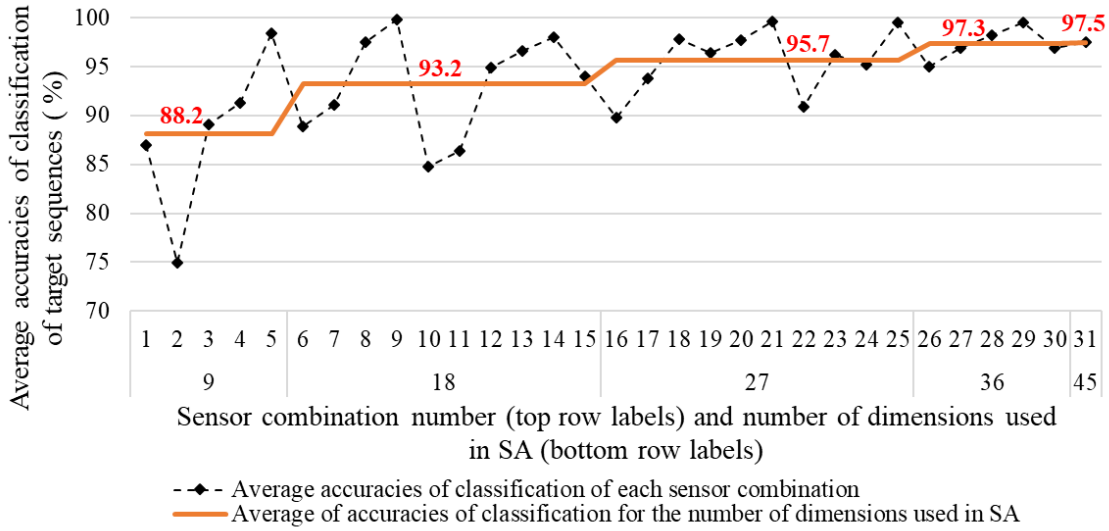


Figure IV-11 Accuracy of classification of target sequence activities with different training samples and SA data from different number of sensors for subject-independent classification

The major conclusion that can be drawn from Figure IV-11 is the fact that changing the combination of sensors even within the same number of dimensions causes variations in the classification accuracy. For example, sensor combinations 9 and 10 use data from 2 sensors each; however, due to the difference in the sensors chosen, the average classification accuracy is 15% less when using data from sensor combination 10 than sensor combination 9 (i.e. it decreases to 85% from 100%). Furthermore, it can be concluded that an increase in the number of sensors generally increases the average classification accuracy. This can be visually confirmed in Figure IV-11 by tracking the gradual upward trend of the bold line. Overall, this amounts to a cumulative increase of

9.3%, from the initial value of 88.2% when using only 1 sensor to a final value of 97.5% when using all 5 sensors.

In order to present findings with greater granularity, the classification accuracy for different training and testing combinations across various sensor combinations is also examined and presented in Figure IV-12. In this Figure, dashed lines represent the classification accuracy obtained for each of the training and testing combinations, whereas the bold solid line shows the average classification accuracy for a given number of sensors, across all training and testing combinations. Similar to the conclusion drawn from the investigation of subject-dependent classification, while a high accuracy is maintained in most combinations, in some cases the classification accuracy is relatively lower. Naturally, due to the variations in how individual subjects perform their activities, the number of combinations with classification accuracy of more than 90% decreases from 212 out of 248 (85% of the combinations) in subject-dependent classification to 188 out of 248 (75% of the combinations) in subject-independent classification. However, as Figure IV-12 indicates, the average accuracy across all the training and testing combinations is above 90% when using 2 or more sensors.

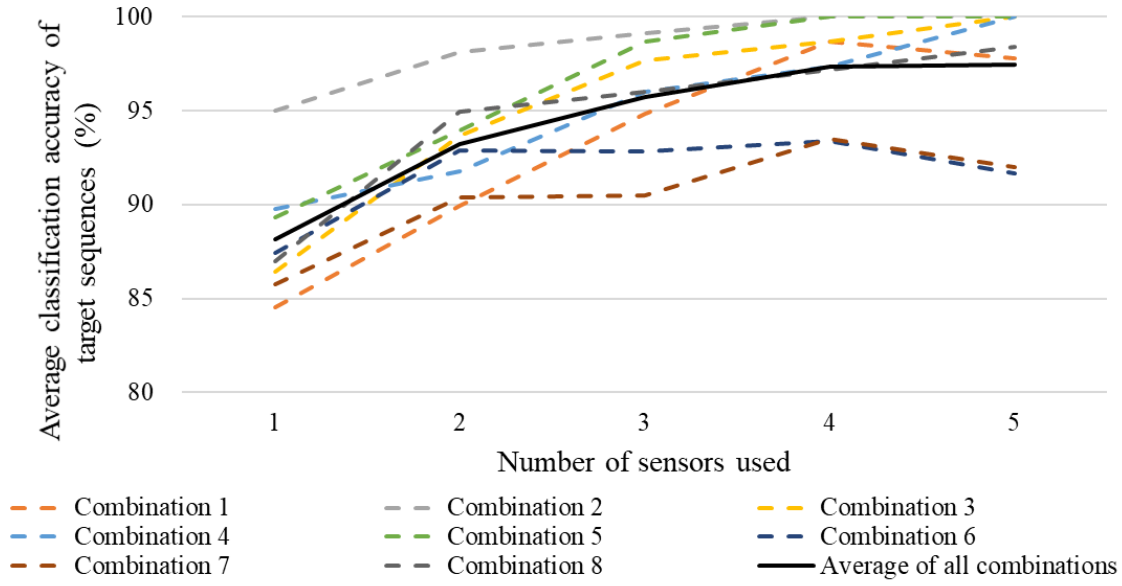


Figure IV-12 Accuracy of classification of target sequence activities with different training samples and over different dimensional SA in subject-independent classification

Finally, as shown in Figure IV-13, the computation time required to perform subject-independent classification increases linearly in relation to the number of dimensions of comparison, while the variation in accuracy (vertical bars in Figure IV-13) is more random. In this Figure, the right vertical axis shows time taken by the algorithm to train 100 minutes of data and subsequently classify 100 minutes of data using the identified parameters. Among all training and testing combinations, the highest accuracy of 97.5% is achieved with 45-dimensional SA.

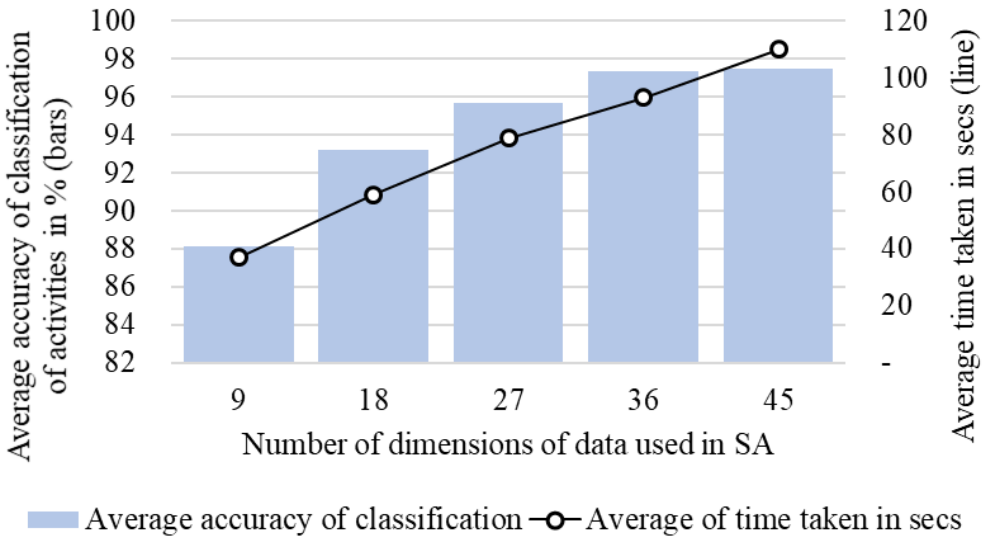


Figure IV-13 Accuracy of classification of target sequence activities and computation time required with different training samples and SA data from different number of sensors for subject-independent classification

IV.4.3 Comparing the effectiveness of MSA and HAR in generating simulation input models

Results obtained from processing raw time-motion sensor data with MSA are then used to assess whether the designed single-step MSA could produce equally or more stable simulation input models compared to the two-step HAR-SA scheme described in Chapter III. For this purpose, a sequential discrete event simulation (DES) model consisting of 5 activities is used, as shown in Figure IV-14. In this Figure, links having a 100% weight value represent clear transition paths between successive activities (i.e. deterministic model). Three copies (A, B, and C) of this model are then created which vary only in how their input models are generated. The simulation input parameters are obtained from the

ground truth information (activity labels as reported in the UCI dataset) in copy A, from the results of the two-step HAR-SA algorithm in copy B, and from the results of the single-step MSA algorithm in copy C. For consistency, identical training and testing combinations are used for both algorithms in each of the 100 sequences tested.

Figure IV-14 Deterministic form of the simulation model (copy A)

In order to run a statistically significant sample, 100 sequences comprising of 25 minutes of activity sensor data is chosen from the testing dataset. The first two target sequences are illustrated in Table IV-3.

Table IV-3 Different activity sequence tested

Sequence no	Activity	1	2	3	4	5
1	Subject	1	4	7	6	5
	Activity	Walking	Walking upstairs	Sitting	Standing	Running
2	Subject	5	7	2	8	2
	Activity	Running	Standing	Sitting	Walking upstairs	Walking

In order to compare the variation in results obtained from models A, B, and C, all three copies are first run several times for each of the 100 sequences of activities. While in copy A, activity transitions are deterministic (Figure IV-14), in copies B and C these transitions are treated as probabilistic where the probabilities depend on extracted activity transition information reported by HAR-SA and MSA algorithms. In Chapter II, it was explained how these transitions are extracted from raw time-motion data, and later used to create a matrix called the dependency network assimilator (DNA) matrix. The incorporation of the extracted DNA into the simulation input model results in a non-deterministic model illustrated in Figure IV-15.

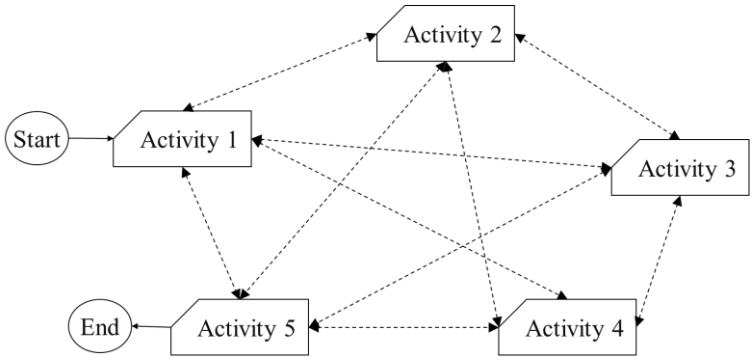


Figure IV-15 Non-deterministic form of the simulation model (copies B and C)

The first cost metric (i.e. objective function) of the simulation model is defined as a measure of the effort required to perform various activities. When an individual performs an activity, energy is consumed by the body to produce work. This energy is commonly measured in terms of calories. By definition, 1 calorie is the amount of energy required to

raise the temperature of 1 gram of water by 1 °C (Hargrove 2006). Naturally, calorie requirement varies with the different activities being performed and the physical traits of the performer such as age, gender, height, and health status. In the context of this research and to have a realistic benchmark for comparing copies A, B, and C of the simulation model, the number of calories expended in performing each activity for a specified duration is used to calculate the overall effort needed to complete a particular activity sequence. Since the specifics of the physical characteristic of the subjects in the UCI dataset are unknown, the calories burned by an average 20-30-year-old person while performing the activities are used to calculate the effort needed to perform those activities. These values are listed in Table IV-4.

Table IV-4 Average calories consumed for various activities (for subjects aging between 20 and 30 years)

Activity	Calorie count per hour	Source
Standing	140	(Buckley et al. 2014)
Walking upstairs	563	(Wisconsin DHHS 2017)
Walking	280	(US DHHS and NIH 2006)
Running	590	(US DHHS and NIH 2006)
Jumping	704	(Wisconsin DHHS 2017)

The second cost metric (i.e. objective function) takes into account the effort required to transition between different activities, as illustrated by Figure IV-16. Values in this Figure are expressed in relative terms, with value 0 as a benchmark. For instance,

while transitioning from an ‘standing’ position to another ‘standing’ position requires zero effort, transitioning from ‘standing’ to ‘walking upstairs’ requires that 4 effort units are consumed. Similarly, a transition from ‘walking upstairs’ to ‘running’ is costlier and requires 12 units of effort. It is worth noting that although the values presented in Figure IV-16 are currently chosen by intuition, they can be linked to factors such as muscle strain and joint fatigue that may result from sudden transition from one activity to another. While exploring and quantification of these relationships is beyond the scope of this Thesis, it can be a potential direction for future work in this area.

	Standing	Walking Upstairs	Walking	Running	Jumping
Standing	0	4	3	5	5
Walking Upstairs	3	0	3	12	16
Walking	3	3	0	10	15
Running	5	7	5	0	25
Jumping	5	8	5	20	0

Figure IV-16 Relative costs (effort) of transition between different activities

IV.4.3.1 Data input modeling for non-deterministic DES model validation

As discussed, two sets of non-deterministic input are generated and used to validate the performance of the designed MSA methodology as applicable to DES input modeling. Here, the efficacy of the algorithm is evaluated using the global fitness parameter (GFP) formulated in Equation III-2 which expresses the ratio of correctly identified activity instances to the total number of activity instances. Using this

convention, the GFP of MSA is calculated as 96.1% (C.I. [95.3% 97.0%], α = 0.05 with σ = 4.3%) with an interquartile range of 6%. Similarly, for HAR-SA, the GFP is obtained as 97.15% (C.I. [96.2% 98.0%], α = 0.05 with σ = 4.45%) with an interquartile range of 5%. It can thus be inferred that the classification accuracy achieved from both methods are similar with a significant overlap in the 95% C.I. of the GFP values. Moreover, the performance of classification using MSA and HAR is evaluated using measures of precision and recall. These measures are more sensitive to the error of classification and incorporate the fact that the cost of misclassification can vary among different scenarios (Nath 2017). Mathematically, precision and recall are expressed by Equation IV-4 and Equation IV-5, respectively, in which TP, FP, and FN indicate true positive, false positive, and false negative instances in activity recognition.

$$Precision = \frac{TP}{TP+FFP} \tag{IV-4}$$

$$Recall = \frac{TP}{TP+FN} \tag{IV-5}$$

To calculate precision and recall values for MSA and HAR, corresponding activity recognition confusion matrices, as illustrated in Figure IV-17 and Figure IV-18, are used.

	Standing	Walking Upstairs	Walking	Running	Jumping
Standing	100%	0%	0%	0%	0%
Walking Upstairs	4%	94%	3%	0%	0%
Walking	0%	3%	97%	0%	0%
Running	0%	0%	0%	100%	0%
Jumping	1%	1%	3%	5%	90%

Figure IV-17 Confusion matrix obtained for classification of 100 testing sequences using MSA

	Standing	Walking Upstairs	Walking	Running	Jumping
Standing	100%	0%	0%	0%	0%
Walking Upstairs	0%	96%	4%	0%	0%
Walking	0%	0%	100%	0%	0%
Running	0%	0%	0%	100%	0%
Jumping	0%	1%	2%	1%	96%

Figure IV-18 Confusion matrix obtained for classification of 100 testing sequences using HAR-SA

The calculated precision and recall values are listed in Table IV-5, which indicates that both HAR-SA and MSA algorithms yield high precision and recall. For instance, the precision and recall of classification is at least 95% and 90%, respectively for all activities in both classification algorithms. Overall, these observations confirm the high reliability of both classification algorithms. Moreover, while both classification algorithms achieve high precision and recall, for some activities, HAR-SA classification achieves marginally better precision and recall than MSA classification. For example, for activity ‘walking

upstairs' the precision of classification using MSA is 95% whereas this value is 99% for HAR-SA. Similarly, the recall of classification for this activity is 94% using MSA classification compared to 96% for HAR-SA.

Table IV-5 Precision and recall of different activities using MSA and HAR-SA

Measure	Classification algorithm	Activities					Weighted average
		Standing	Walking Upstairs	Walking	Running	Jumping	
Precision	MSA	95%	95%	95%	95%	100%	96%
	HAR	100%	99%	95%	99%	100%	98%
Recall	MSA	100%	94%	97%	100%	90%	96%
	HAR	100%	96%	100%	100%	96%	98%

Next, the extracted DNA matrix and the average duration of each activity for both classification methods are derived for each of the target sequences, and this information is used as input for generating the non-deterministic DES models (copies B and C).

IV.4.3.2 Analysis of the output of the non-deterministic DES model

All three copies of the simulation model (copy A generated from the ground truth, copy B from HAR-SA, and copy C from MSA) are then launched for each of the 100 sequences. In analyzing the results, the total cost and the transition cost derived from model A is regarded as benchmark, with the total costs and transition costs derived from models B and C being compared against this benchmark. The percentage difference in the total cost from model A to the total cost derived from models B and C for each of the 100

sequences is shown in Figure IV-19, and the percentage difference in the transition costs derived from model A to the transition cost derived from model B and C for each of the 100 sequences is shown in Figure IV-20.

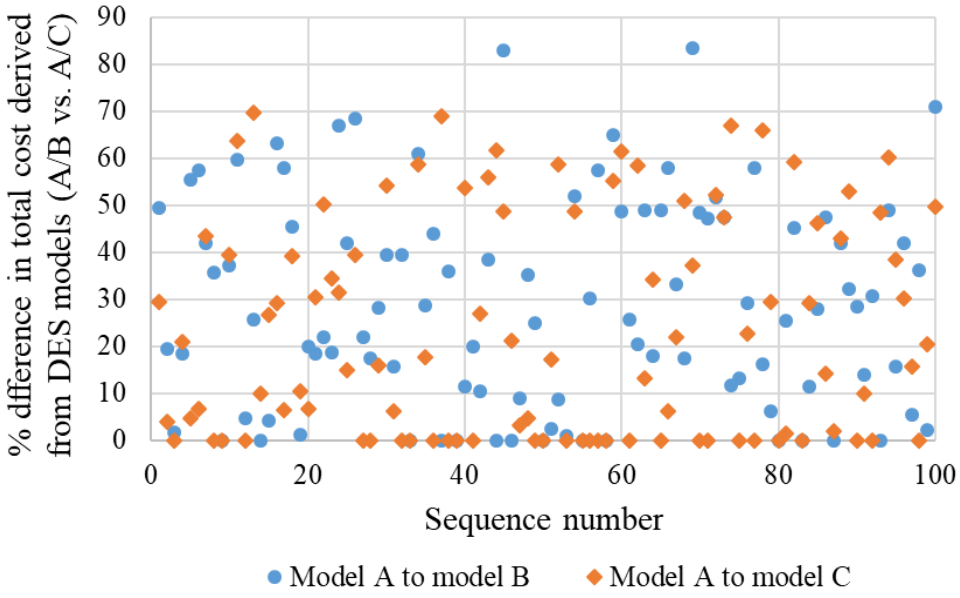


Figure IV-19 % difference in the total cost derived from model A (ground truth) to models B (HAR-SA) and C (MSA)

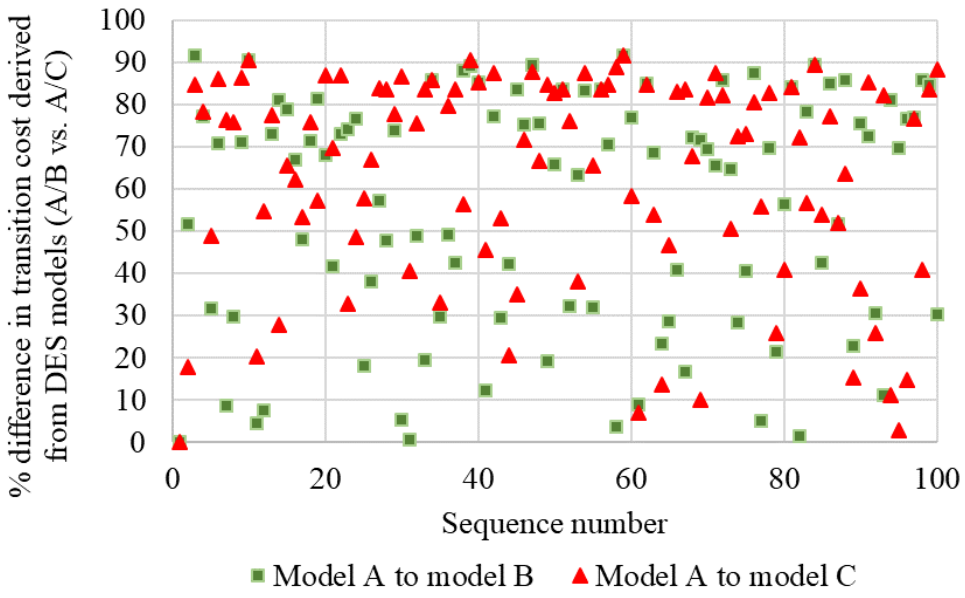


Figure IV-20 % difference in the transition cost derived from model A (ground truth) to models B (HAR-SA) and C (MSA)

Further analysis reveals that on average the percentage discrepancy in the total cost derived from model C (MSA) was 24% (C.I. [19.2% 28.4%], α = 0.05 with σ = 23.2%) of the total cost derived from model A (ground truth), whereas the same ratio was 29% (C.I. [24.3% 33.1%], α = 0.05 with σ = 22.1%) between model B (HAR-SA) and model A. Moreover, the percentage discrepancy in the transition cost derived from model C (MSA) was 63% (C.I. [58.4% 68.3%], α = 0.05 with σ = 24.8%) of the total cost derived from model A (ground truth), whereas the same ratio was 56% (C.I. [50.5% 61.6%], α = 0.05 with σ = 27.8%) between model B (HAR-SA) and model A. The overlap in 95% C.I.'s of the mean discrepancy between the costs derived from Model A and Model B, and Model A and Model C suggest that, overall both algorithms have similar efficacy of

classification. Moreover, in the terms of the percentage discrepancy of total cost from the ground truth, model C yielded closer results in 53 out of the 100 sequences, whereas in terms of discrepancy of transition costs from the ground truth model C yielded closer results in only 39 of the 100 sequences. These observations provide a statistical validation of the hypothesis that using the performance of MSA to generate simulation input models can match (if not exceed) the performance of HAR-SA.

IV.5 Summary and conclusion

In this Chapter, the MSA technique was explored as a possible bridge between raw sensor data and simulation model inputs. In a nutshell, MSA operates by comparing various attributes of sequences of multi-dimensional data and forming conclusions by aggregating the scores obtained from the various comparisons to produce an activity classification with high fidelity. In comparison with traditional HAR techniques (as described in Chapter II), MSA allows for more flexibility as it (i) considers trends between individual data points, (ii) is not limited to specific window sizes for activity recognition, and (iii) can be applied to sequence of activities at once while incorporating key information about underlying activity dependencies among others.

An exploration of HAR and MSA operations showed that while both rely on simplification of the available data by deriving representatives for particular subsets of the data, HAR uses statistical features whereas MSA is based on categorical representation.

The derivation of categorically representing continuous sensor data is a widely validated approach, and commonly used in supervised ML applications.

The designed MSA algorithm was systematically examined as an alternative to the previously developed HAR-SA framework. It was found that MSA not only did reduce a two-step activity classification process to a single step implementation, it also expanded the possibility of using prior knowledge describing the sequences of and relationships among activities. It was further concluded that while MSA can be used as a substitute to HAR for pre-processing of ergonomic (time-motion) data, it can be also incorporated with HAR to improve the overall activity classification performance in the presence of non-ergonomic data. For example, sequences of activities identified via HAR can be further aligned with the available data on cost and quality to identify anomalies in the classification results.

The developed methodology in this Chapter followed an integrated process of data categorization, determination of parameters through cross validation using available labeled data and new unlabeled sequences. This framework was implemented for a publicly available dataset with the objective of correctly classifying the activities from the raw sensor data. The dataset consisted of 8 subjects performing 5 activities for 5 minutes each while data was collected by 5 sensor units in a total of 45 dimensions.

The MSA algorithm was applied to both subject-dependent and subject-independent scenarios. After several iterations, the average accuracy of activity recognition was found to be 98% and 97% for subject-dependent and subject-independent

classification, respectively, when all 45 dimensions were used. The output of the subject-independent activity recognition was then used as input for a non-deterministic DES model. The same model was also built using output from HAR-SA algorithm. Simulation outputs (measured in terms of the calories burned and the energy expended in transition between activities) form these two inputs were compared against the simulation output using the ground truth in order to establish the validity of MSA as a means to generate reliable simulation input models. Results showed that in 53 out of 100 sequences the output of MSA outperformed the output of HAR-SA in terms of total cost, while this number was 39 out of 100 for activity transition cost.

These findings confirm that MSA is a viable alternative to HAR-SA while expanding the scope of prior information. While the current implementation focusses on recognizing the patterns within the activities, it can be also adapted for recognizing patterns across sequences. For instance, when data describing different attributes in different instances is available, MSA can be used to compare across those instances. Moreover, input data may be ordinal as well as continuous. For example, if data on costs, activities, and schedule is available, a 3-dimensional SA can be adopted and used to classify and compare activity sequences and identify anomalies.

It must be noted that the current implementation of MSA is limited to an alphabet of 20 characters due to the fact that sequence alignment applications were developed primarily for bioinformatics applications which require representation of only the 20 basic amino acids that make up most of biological matter. Future work in this area will mainly

focus on expanding the range of categories beyond the existing 20 characters. It is expected that with greater granularity in the categorical information, the accuracy can be improved thus providing greater stability of resulting simulation input models.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

V.1 Conclusions

The general advancement in technology over the past few years has created significant opportunities of efficiency improvement in the construction industry, as one of the most important sectors of the global and U.S. economy. Such improvement is expected to enable the industry to shed its traditional mantle as it slowly adopts to technological advances to overcome current stagnant low productivity rates. In particular, with the proliferation of data, data-driven discrete event simulation (DES) modeling has been proposed as a potentially effective platform to examine the uncertainties in project planning and execution, and accelerate the adoption of data-driven decision-making during project lifecycle. However, despite the availability of large and diverse volumes of data, the integration of data-enabled techniques such as data-driven simulation has been hindered due to the presence of noise in the collected input data, which in turn deteriorates the reliability and fidelity of simulation output. Moreover, there is a lack of comprehensive frameworks that can process raw process-level data to increase the general reliability and stability of model outputs.

In light of these fundamental challenges, the work presented in this Thesis aimed at filling existing gaps in knowledge and practice by examining the hypothesis that techniques derived from key natural phenomena that deal with noise can improve the

quality of input modeling in DES systems. This was objectively evaluated using comparisons in the context of resemblance of simulation output to ground truth information, ability to factor in domain-specific parameters and constraints, or a combination of these measures. In particular, two major categories of natural phenomena were investigated in this Thesis; evolutionary techniques, also known as genetic algorithms (GA), and sequence alignment (SA) (both pairwise and multi-dimensional). Work presented in this Thesis validated the central hypothesis through improvement in the resemblance to results from ground truth. In the discussion that follows, the validation of the hypothesis using each of the three major techniques (GA, SA and MSA) is summarized.

Chapter II dealt with the improvement of activity recognition transition data by using GA, an evolutionary technique with roots in nature. In the framework illustrated in this Chapter, human time-motion data was collected from a warehouse operation experiment using built-in smartphone sensors (accelerometer, linear accelerometer, and gyroscope). Collected sensor readings were first processed through a machine learning (ML) framework, focused on human activity recognition (HAR) algorithms. The output of HAR was then used to extract and refine activity transition information, and document this information in a dependency network assimilator (DNA) matrix form. The fitness of the generated DNA matrix was improved in an iterative GA-DES process from 0.76 to 0.96 (compared to the ground truth value of 0.97). This improvement was further validated by using the obtained activity transition information as input of a simulation model, and

assessing the quality of output in terms of the total time used to inspect and process each box, variation in the unit cost, and inspector's idle time. This increased resemblance of the simulation results to the ground truth provided clearer insight into the real system, potentially improving the quality of decision-making with regards to safety and health, ergonomics, resource allocation, and jobsite layout.

The work presented in Chapter III expanded the refinement of activity transition data laid out in Chapter II through the use of SA. This Chapter primarily dealt with the errors in the activity recognition output of HAR and used the principles used in the comparison of deoxyribose nucleic acid (DNA) sequences in bioinformatics to detect and correct potential anomalies in activity sequences. In particular, the global fitness parameter (GFP) which expresses the overall accuracy of activity recognition improved from 85.7% before the implementation of SA to 87.25% after the implementation of SA.

The algorithms presented in Chapter II and Chapter III helped make improvements in the quality of the raw data available as simulation input. However, both were limited to improving the output obtained from classic HAR algorithms, thus necessitating a two-step process (i.e. HAR, followed by either GA or SA). To eliminate this need, the potential of MSA in recognizing activities directly from raw sensor data (without the need of running HAR algorithms) was explored in Chapter IV. MSA expands upon the principles of SA by simultaneously comparing data from several attributes. Like HAR, this algorithm simplifies continuous data, however, unlike HAR, MSA relies on categorical representation of data as opposed to statistical features. The designed MSA algorithm was

validated by implementing it in an openly available human time-motion dataset which contained information collected from 5 sensors (mounted on different body parts) and 8 subjects (both male and female). After several iterations, the average accuracy for subject-dependent activity recognition (for 45-dimensional SA) was calculated as 98%, compared to 97% in subject-independent classification. Further validation was derived by using the outputs of MSA and HAR-SA to generate two separate input models for a 5-activity DES model, and testing the resemblance of simulation results to the ground truth. Results showed that in terms of total cost, in 53 out of 100 sequences MSA outperformed HAR-SA, while in 39 out of 100 sequences MSA outperformed HAR when activity transition cost was considered as a metric. These findings revealed that MSA is a viable alternative to HAR-SA while expanding the scope of prior information.

Overall, the work presented in this Thesis contributes to the body of knowledge and practice by introducing and validating a general framework of sensor data processing inspired by natural phenomena. The algorithms designed and implemented in this Thesis not only do expand the scope of data processing in construction applications but can collectively facilitate a paradigm shift from computationally intensive synthetic data processing techniques to more robust methods of noise refinement in large datasets with built-in dependencies among individual data points.

At a practical level, these methodologies facilitate and ultimately automate the tedious process of collecting, processing, and integrating time-stamped human motion data, for easier and more reliable recognition of performed activities. In achieving this

improvement some priori knowledge is assumed about the system to facilitate greater understanding of the situation. Incorporating certain priori knowledge to produce more insightful posteriori knowledge is a well utilized technique in construction research. For instance, in order to discover process-level knowledge of construction activities, Akhavian and Behzadan (2018), assumed knowledge on the number of subjects, the activities performed, the number of work cycles, and the operational dependencies between the different activities. Similarly, in a system developed with the objective of improving safety performance by detecting and documenting near-miss falls in construction sites using semi-supervised learning algorithms, Yang et. al. (2016) assumed that the number of subjects, the dimensions of the steel frame, the duration of the activity and the start and end timestamps were known. Furthermore, in order to enable more efficient work sampling, Joshua and Varghese (2010) implemented an automated activity recognition system that utilized priori knowledge such as the components of the bricklaying activity performed and the number of subjects.

Similar priori knowledge was assumed in the algorithms implemented in this Thesis as well. In particular, in the implementation of GA (Chapter II), it was assumed that in the modeled system 8 activities were performed, the transition from any given activity to succeeding activities could be determined probabilistically or deterministically (e.g. the inspection station node was a probabilistic node, whereas other nodes were deterministic as shown in the DNA matrix of Figure II-6), the experiment was repeated for 30 cycles, and that each activity was discrete and not immediately followed by another

instance of the same activity. Similar assumptions were made in the implementation of SA as well (Chapter III). For instance, it was assumed that 6 activities were performed for 6 cycles by 4 subjects. Moreover, the ground truth of the first cycle was assumed to be known, while the ground truth of the other cycles remained unknown and random. Likewise, the assumptions made during the implementation of the MSA (Chapter IV) involved knowing that there were 5 activities, performed by 8 subjects with the data collected by 5 sensor units in 45 dimensions or attributes.

In the methodologies developed and implemented in this Thesis, the assumptions that were made and utilized provided a foundational understanding of the framework of the system involved: the specifics were understood only after the implementation of the pre-processing, post-processing, and simulation models for several iterations. The conclusions drawn from these frameworks provided the foundation for data-driven decision-making. For instance, results of the GA enabled recognition of the actual transitions between activities (shown in the DNA matrix of Figure II-13), and more accurate assessment of the cost and time required to move a certain number of boxes in the system (shown in Figure II-16 and Figure II-17). Similarly, incorporating data refinement (pre- and post-processing) steps inspired by SA (Section III.5) and MSA (Section IV.4) techniques increased the understanding of the system though increased accuracy of activity recognition in the system within the context of other information available about the system. In particular, in SA, the sequence of activities in cycle 1 was evaluated to provide insights about the rest of the system, and in MSA, the sensor

information available from the source sequences was used to recognize unknown sequences.

In general, the contributions of SA and MSA were not limited to the specific system being evaluated, as the output can be used to assess other systems as well. For example, the process laid out in Sub-section III.4.2 to assess the reliability of HAR in identifying a particular activity can be valuable in other applications involving similar activities. Taking this further, collection of similar observations can also help improve the heuristics generally used in data science. For instance, let us assume the commonly used value for one heuristic is 10 seconds but repeated simulations through data collected in experiments and project settings reveal the heuristic to perform better with a different value, for example, 8.8 seconds. Moreover, while the current implementation of MSA focusses on recognizing the patterns within activities, it can be easily adapted to be utilized in recognizing patterns across sequences. For instance, given data on different attributes in different instances, the information can be used to compare across those instances. In fact, this comparison in many ways is simpler as available data can be ordinal as well as continuous. For example, since data on cost, activities, and time (schedule) in many projects are available, a 3-dimensional SA can be conveniently used to compare the sequences and identify anomalies. Overall, this alternative approach to pre-processing and post-processing increases the richness of the conclusions derived. Ultimately this will contribute to the generation of more realistic inputs for simulation modeling of real world operations, thus supporting the prospect of data-driven decision-making.

V.2 Directions for future work

In the future, the nature-inspired methods presented in this Thesis will be expanded and improved to ensure easier implementation in the field while producing greater value thorough simulations.

In particular, with further research the use of the coupled GA-DES framework will be expanded to cover more sophisticated scenarios where larger numbers of entities interact in more complex settings beyond the controlled experimental scenario used in this Thesis. Specifically, the number of decision points (i.e. forks) will be expanded to increase resemblance to real construction scenarios. Moreover, further research will minimize the issues that arise in scaling this algorithm to greater scope and complexity.

In addition, the scope of MSA will be expanded beyond the current implementation to enable the recognition of more diverse, multi-attribute activity sequences with better computational efficiency. In essence, the algorithm will be utilized to recognize patterns across sequences with numerical and non-numerical attributes. For example, given project-specific data such as cost, schedule, and activity dependencies, a 3-dimensional alignment can be used to compare the sequences and identify the anomalies. Furthermore, the current implementation of MSA is limited to an alphabet of 20 characters, which places an artificial constraint on the implementation. Thus, future work will focus on expanding the range of categories beyond the current limit of 20. It is expected that with greater granularity in the categorical information, the accuracy is improved while computational requirements are reduced, thus improving the overall

stability of resulting simulation input models. Overall, these improvements are expected to facilitate greater degree of near-real time feedback minimizing the time lapse between data collection and data-driven decision-making.

REFERENCES

Abbott, A. (1995). "Sequence analysis: New methods for old ideas." *Annual Review of Sociology*, 21(1), 93–113.

Abbott, A., and Forrest, J. (1986). "Optimal Matching Methods for Historical Sequences." *Journal of Interdisciplinary History*, 16(3), 471.

Abbott, A., and Tsay, A. (2000). "Sequence analysis and optimal matching methods in sociology: Review and prospect." *Sociological Methods & Research*, 29(1), 3–33.

AbouRizk, S., Halpin, D., Mohamed, Y., and Hermann, U. (2011). "Research in Modeling and Simulation for Improving Construction Engineering Operations." *Journal of Construction Engineering and Management*, 137(10), 843–852.

Akhavian, R. (2015). "Data-driven simulation modeling of construction and infrastructure operations using process knowledge discovery." PhD. Thesis, University of Central Florida, Orlando, FL.

Akhavian, R., and Behzadan, A. H. (2013a). "Knowledge-based simulation modeling of construction fleet operations using multimodal-process data mining." *Journal of Construction Engineering and Management*, 139(11), 04013021.

Akhavian, R., and Behzadan, A. H. (2013b). "Design requirements of an automated data-driven simulation model generator for construction operations."

Proceedings of the International Conference on Civil and Building Engineering Informatics (ICCBEI), Koto, Japan, 114–124.

Akhavian, R., and Behzadan, A. H. (2015). "Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers." *Advanced Engineering Informatics*, 29(4), 867–877.

Akhavian, R., and Behzadan, A. H. (2016). "Smartphone-based construction workers' activity recognition and classification." *Automation in Construction*, 71, 198–209.

Akhavian, R., and Behzadan, A. H. (2018). "Coupling human activity recognition and wearable sensors for data-driven construction simulation." *Journal of Information Technology in Construction (ITcon)*, 23(1), 1–15.

Akhavian, R., Brito, L., and Behzadan, A. (2015). "Integrated mobile sensor-based activity recognition of construction equipment and human crews." *Proceedings of the 2015 Conference on Autonomous and Robotic Construction of Infrastructure*, Ames, IA, 1–21.

Assaf, S. A., Al-Khalil, M., and Al-Hazmi, M. (1995). "Causes of delay in large building construction project." *Journal of Management in Engineering*, 11(2), 45–50.

Azhar, S., Nadeem, A., Mok, J., and Leung, B. (2008). "Building Information Modeling (BIM): A new paradigm for visual interactive modeling and simulation for

construction projects." *Proceedings of the First International Conference on Construction in Developing Countries*, Karachi, Pakistan, 435–446.

Banks, J. (1998). *Handbook of simulation: principles, methodology, advances, applications, and practice*. John Wiley & Sons, Hoboken, NJ.

Barshan, B., and Yüksek, M. C. (2014). "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units." *The Computer Journal*, 57(11), 1649–1667.

Barzilay, R., and Lee, L. (2003). "Learning to paraphrase: an unsupervised approach using multiple-sequence alignment." *In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Association for Computational Linguistics, Edmonton, Canada, 16–23.

Bathula, M., Ramezanali, M., Pradhan, I., Patel, N., Gotschall, J., and Sridhar, N. (2009). "A sensor network system for measuring traffic in short-term construction work zones." *Distributed Computing in Sensor Systems: 5th IEEE International Conference, DCOSS 2009, Marina del Rey, CA, USA, June 8-10, 2009. Proceedings*, Springer Berlin Heidelberg, Berlin, Heidelberg, 216–230.

Becerik-Gerber, B., Gerber, D. J., and Ku, K. (2011). "The pace of technological innovation in architecture, engineering, and construction education: integrating

recent trends into the curricula." *Journal of Information Technology in Construction (ITcon)*, 16(24), 411–432.

Becerik-Gerber, B., Siddiqui, M. K., Brilakis, I., El-Anwar, O., El-Gohary, N., Mahfouz, T., Jog, G. M., Li, S., and Kandil, A. A. (2013). "Civil engineering grand challenges: Opportunities for data sensing, information analysis, and knowledge discovery." *Journal of Computing in Civil Engineering*, 28(4), 04014013.

Blair-Loy, M. (1999). "Career patterns of executive women in finance: An optimal matching analysis." *American Journal of Sociology*, 104(5), 1346–1397.

Blasch, E., Seetharaman, G., and Reinhardt, K. (2013). "Dynamic data driven applications system concept for information fusion." *Procedia Computer Science*, 18, 1999–2007.

Buckley, J. P., Mellor, D. D., Morris, M., and Joseph, F. (2014). "Standing-based office work shows encouraging signs of attenuating post-prandial glycaemic excursion." *Occupational and Environmental Medicine*, 71(2), 109–111.

Bureau of Labor Statistics (BLS). (2015). "Occupational employment and wages." (http://www.bls.gov/oes/current/oes472061.htm) (Mar. 2, 2017).

Carr, R. I. (1979). "Simulation of construction project duration." *Journal of the Construction Division*, 105(2), 117–128.

Chae, M. J., Yoo, H. S., Kim, J. Y., and Cho, M. Y. (2012). “Development of a wireless sensor network system for suspension bridge health monitoring.” *Automation in Construction*, 21, 237–252.

Chan, W.-T., Chua, D. K., and Kannan, G. (1996). “Construction resource scheduling with genetic algorithms.” *Journal of Construction Engineering and Management*, 122(2), 125–132.

Chen, L., and Khalil, I. (2011). “Activity recognition: Approaches, practices and trends.” *Activity Recognition in Pervasive Intelligent Environments*, Springer, 1–31.

Chmielewski, M. R., and Grzymala-Busse, J. W. (1996). “Global discretization of continuous attributes as preprocessing for machine learning.” *International journal of approximate reasoning*, 15(4), 319–331.

Choe, S., Leite, F., Seedah, D., and Caldas, C. (2014). “Evaluation of sensing technology for the prevention of backover accidents in construction work zones.” *Journal of Information Technology in Construction (ITcon)*, 19(1), 1–19.

Colubi, A., and González-Rodríguez, G. (2015). “Fuzziness in data analysis: Towards accuracy and robustness.” *Fuzzy Sets and Systems*, 281, 260–271.

Copasaro, G. (2018). “An introduction to applied bioinformatics.” (http://readiab.org/book/0.1.3/) (Jan. 22, 2018).

CPWR. (2016). “Workplace Safety and Health Perceptions of Construction Workers.” (http://www.cpwr.com/publications/third-quarter-workplace-safety-and-health-perceptions-construction-workers) (Dec. 25, 2017).

Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, NY.

Dougherty, J., Kohavi, R., and Sahami, M. (1995). “Supervised and unsupervised discretization of continuous features.” *Machine Learning Proceedings 1995*, Elsevier, 194–202.

Dunham, M. H. (2006). *Data mining: Introductory and advanced concepts*. Pearson Education, Upper Saddle River, NJ.

Edgar, R. C. (2004). “MUSCLE: a multiple sequence alignment method with reduced time and space complexity.” *BMC bioinformatics*, 5(1), 113.

Elias, I. (2006). “Settling the intractability of multiple alignment.” *Journal of Computational Biology*, 13(7), 1323–1339.

Estrin, D., Borriello, G., Colwell, R., Fiddler, J., Horowitz, M., Kaiser, W., Leveson, N., Liskov, B., Lucas, P., and Maher, D. (2001). *Embedded, everywhere: A research agenda for networked systems of embedded computers*. National Research Council, Washington, DC.

Flores, J. J., Antolino, A., and Garcia, J. M. (2009). “Evolving hidden markov models for network anomaly detection.” *Proceedings of the 8th Mexican International Conference on Artificial Intelligence,(MICAI 2009),* Guanajuato, Mexico.

Gao, T., Ergan, S., Akinci, B., and Garrett, J. H. (2013). “Proactive productivity management at job sites: Understanding characteristics of assumptions made for construction processes during planning based on case studies and interviews.” *Journal of Construction Engineering and Management*, 140(3), 04013054.

Gleick, J. (1987). *Chaos: Making a New Science*. Open Road Media, New York, NY.

Golparvar-Fard, M., Heydarian, A., and Niebles, J. C. (2013). “Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers.” *Advanced Engineering Informatics*, 27(4), 652–663.

Golparvar-Fard, M., Peña-Mora, F., and Savarese, S. (2011). “Integrated sequential as-built and as-planned representation with 4D AR tools in support of decision-making tasks in the aec/fm industry.” *Journal of Construction Engineering and Management*, 137(12), 1099–1116.

Hajjar, D., and AbouRizk, S. M. (2002). “Unified modeling methodology for construction simulation.” *Journal of Construction Engineering and Management*, 128(2), 174–185.

Han, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier, Burlington, MA.

Hargrove, J. L. (2006). "History of the Calorie in Nutrition." *The Journal of Nutrition*, 136(12), 2957–2961.

Harrington, P. (2012). *Machine learning in action*. Manning, Shelter Island, NY.

Higgins, D. G., and Sharp, P. M. (1988). "CLUSTAL: A package for performing multiple sequence alignment on a microcomputer." *Gene*, 73(1), 237–244.

HPRC. (2018). "Introduction to ADA." (https://hprc.tamu.edu/wiki/Ada:Intro#Hardware_Summary) (Feb. 7, 2018).

Huang, P.-C., Lee, S.-S., Kuo, Y.-H., and Lee, K.-R. (2010). "A flexible sequence alignment approach on pattern mining and matching for human activity recognition." *Expert Systems with Applications*, 37(1), 298–306.

Huang, Y., and Verbraeck, A. (2009). "A dynamic data-driven approach for rail transport system simulation." *Proceedings of the 2009 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Inc, Austin, TX, 2553–2562.

Islam, M. M., Hassan, M. M., Lee, G.-W., and Huh, E.-N. (2012). "A survey on virtualization of wireless sensor networks." *Sensors*, 12(12), 2175–2207.

Izadi, D., Abawajy, J. H., Ghanavati, S., and Herawan, T. (2015). “A Data Fusion Method in Wireless Sensor Networks.” *Sensors*, 15(2), 2964–2979.

Jang, W.-S., and Skibniewski, M. J. (2009). “Cost-benefit analysis of embedded sensor system for construction materials tracking.” *Journal of Construction Engineering and Management*, 135(5), 378–386.

Joh, C.-H., Arentze, T., Hofman, F., and Timmermans, H. (2002). “Activity pattern similarity: a multi-dimensional sequence alignment method.” *Transportation Research Part B: Methodological*, 36(5), 385–403.

Joshua, L., and Varghese, K. (2010). “Accelerometer-based activity recognition in construction.” *Journal of Computing in Civil Engineering*, 25(5), 370–379.

Kataoka, Y., Nakashika, T., Aihara, R., Takiguchi, T., and Ariki, Y. (2016). “Selection of an optimum random matrix using a genetic algorithm for acoustic feature extraction.” *Proceedings of the 2016 Conference on Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference*, IEEE, Okayama, Japan, 1–6.

Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013). “Multi-sensor data fusion: A review of the state-of-the-art.” *Information Fusion*, 14(1), 28–44.

Kiel, L. D., and Elliott, E. W. (1996). *Chaos theory in the social sciences: Foundations and applications*. University of Michigan Press, Ann Arbor, MI.

Kiziltas, S., Akinci, B., Ergen, E., Tang, P., and Gordon, C. (2008). "Technological assessment and process implications of field data capture technologies for construction and facility/infrastructure management." *Journal of Information Technology in Construction (ITcon)*, 13(10), 134–154.

Kopsida, M., Brilakis, I., and Vela, P. A. (2015). "A review of automated construction progress monitoring and inspection methods." *Proceedings of the 32nd CIB W78 Conference, Endhoven, Netherlands*, 421–431.

Kotsiantis, S., Kanellopoulos, D., and Pintelas, P. (2006). "Data preprocessing for supervised leaning." *International Journal of Computer Science*, 1(2), 111–117.

KPMG. (2015). "Global Construction Survey." (https://assets.kpmg.com/content/dam/kpmg/pdf/2015/04/global-construction-survey-2015.pdf) (Jan. 3, 2018).

Law, A. M., Kelton, W. D., and Kelton, W. D. (1991). *Simulation modeling and analysis*. McGraw-Hill, New York, NY.

Lee, S., Behzadan, A., Kandil, A., and Mohamed, Y. (2013). "Grand challenges in simulation for the architecture, engineering, construction, and facility management industries." *Computing in Civil Engineering (2013)*, 773–785.

Leite, F., Cho, Y., Behzadan, A. H., Lee, S., Choe, S., Fang, Y., Akhavian, R., and Hwang, S. (2016). "Visualization, information modeling, and simulation: Grand

challenges in the construction industry." *Journal of Computing in Civil Engineering*, 30(6), 04016035.

Levy, D. (1994). "Chaos theory and strategy: Theory, application, and managerial implications." *Strategic management journal*, 167–178.

Lichman, M. (2013). "UCI Machine Learning Repository." (http://archive.ics.uci.edu/ml) (Jan. 31, 2018).

Lin, F., and Ying, H. (2002). "Modeling and control of fuzzy discrete event systems." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 32(4), 408–415.

Lin, S.-Y., Chao, K.-M., and Lo, C.-C. (2010). "Service-oriented dynamic data driven application systems to traffic signal control." IEEE, 3463–3470.

Liu, H., Hussain, F., Tan, C. L., and Dash, M. (2002). "Discretization: An enabling technique." *Data mining and knowledge discovery*, 6(4), 393–423.

Liz, E., and Ruiz-Herrera, A. (2012). "Chaos in discrete structured population models." *SIAM Journal on Applied Dynamical Systems*, 11(4), 1200–1214.

Lorenz, E. N. (1963). "Deterministic non-periodic flow." *Journal of the Atmospheric Sciences*, 20(2), 130–141.

Martinez, J. C. (1996). *Stroboscope: State and Resource Based Simulation of Construction Processes*. University of Michigan, Ann Arbor, MI.

Martinez, J. C., and Ioannou, P. G. (1994). “General purpose simulation with stroboscope.” Society for Computer Simulation International, 1159–1166.

Martinez, J. C., and Ioannou, P. G. (1997). “State-based probabilistic scheduling using STROBOSCOPE’s CPM add-on.” *Proceedings of the 1997 Construction Congress V, ASCE, Stuart D. Anderson, edition*, Minneapolis, Minnesota, 438–445.

Marzouk, M., and Moselhi, O. (2003). “Object-oriented simulation model for earthmoving operations.” *Journal of Construction Engineering and Management*, 129(2), 173–181.

Nath, N. (2017). “Construction ergonomic risk and productivity assessment using mobile technology and machine learning.” M.S. Thesis Missouri State University, Springfield, MO.

Needleman, S. B., and Wunsch, C. D. (1970). “A general method applicable to the search for similarities in the amino acid sequence of two proteins.” *Journal of Molecular Biology*, 48(3), 443–453.

Oloufa, A. A., Ikeda, M., and Nguyen, T.-H. (1998). “Resource-based simulation libraries for construction.” *Automation in construction*, 7(4), 315–326.

Park, C.-S., Lee, D.-Y., Kwon, O.-S., and Wang, X. (2013). “A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template.” *Automation in Construction*, 33, 61–71.

Park, H.-S. (2006). “Conceptual framework of construction productivity estimation.” *KSCE Journal of Civil Engineering*, 10(5), 311–317.

Pfahringer, B. (1995). “Compression-Based Discretization of Continuous Attributes.” *Machine Learning Proceedings 1995*, Elsevier, 456–463.

Poli, R., Langdon, W. B., McPhee, N. F., and Koza, J. R. (2008). *A field guide to genetic programming*. Lulu.com.

Polyanovsky, V. O., Roytberg, M. A., and Tumanyan, V. G. (2011). “Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences.” *Algorithms for Molecular Biology*, 6(1), 25.

Portas, J., and AbouRizk, S. (1997). “Neural network model for estimating construction productivity.” *Journal of Construction Engineering and Management*, 123(4), 399–410.

Preis, A., and Ostfeld, A. (2008). “Genetic algorithm for contaminant source characterization using imperfect sensors.” *Civil Engineering and Environmental Systems*, 25(1), 29–39.

Razavi, S. N., and Haas, C. T. (2010). “Multi-sensor data fusion for on-site materials tracking in construction.” *Automation in Construction*, 19(8), 1037–1046.

RazaviAlavi, S., and AbouRizk, S. (2016). “Genetic algorithm–simulation framework for decision-making in construction site layout planning.” *Journal of Construction Engineering and Management*, 143(1), 04016084.

Reeves, C. (2003). “Genetic algorithms.” *Handbook of metaheuristics*, Springer, Berlin, Germany, 55–82.

Ritter, T., Euler, J., Ulbrich, S., and von Stryk, O. (2016). “Decentralized dynamic data-driven monitoring of atmospheric dispersion processes.” *Procedia Computer Sci.*, 80, 919–930.

Rosenberg, M. S. (2009). *Sequence alignment: methods, models, concepts, and strategies*. University of California Press, Berkley, CA.

Rosenfeld, Y. (2014). “Root-cause analysis of construction-cost overruns.” *Journal of Construction Engineering and Management*, 140(1), 04013039.

Sankoff, D., and Kruskal, J. B. (1983). “Time warps, string edits, and macromolecules: the theory and practice of sequence comparison.” *Reading: Addison-Wesley Publication, 1983, edited by Sankoff, David; Kruskal, Joseph B.*

Shoval, N., and Isaacson, M. (2007). “Sequence alignment as a method for human activity analysis in space and time.” *Annals of the Association of American geographers*, 97(2), 282–297.

Shrestha, P., and Behzadan, A. H. (2017). “An evolutionary method to refine imperfect sensor data for construction simulation.” *Proceedings of the 2017 Winter Simulation Conference*, IEEE, Las Vegas, NV.

Shrestha, P., Nath, N. D., and Behzadan, A. H. (2018). “Coupling Machine Learning and Sequence Alignment for Improved Human Activity Recognition from Mobile

Sensor Data." *Proceedings of the 2018 Construction Research Congress (CRC)*, New Orleans, LA.

Simoni, R. D., Hill, R. L., and Vaughan, M. (2002). "The discovery of the amino acid threonine: The work of William C. Rose." *The Journal of Biological Chemistry*, 277(37), E25.

Skoogh, A., Johansson, B., and Stahre, J. (2012). "Automated input data management: evaluation of a concept for reduced time consumption in discrete event simulation." *Simulation*, 88(11), 1279–1293.

Smith, T. F., and Waterman, M. S. (1981). "Identification of common molecular subsequences." *Journal of Molecular Biology*, 147(1), 195–197.

Song, L., and Eldin, N. N. (2012). "Adaptive real-time tracking and simulation of heavy construction operations for look-ahead scheduling." *Automation in Construction*, 27.

Spencer Jr, B., Ruiz-Sandoval, M. E., and Kurata, N. (2004). "Smart sensing technology: Opportunities and challenges." *Journal of Structural Control and Health Monitoring*, 11(4), 349–368.

Srinivas, M., and Patnaik, L. M. (1994). "Genetic algorithms: A survey." *Computer*, 27(6), 17–26.

Tannock, J., Cao, B., Farr, R., and Byrne, M. (2007). “Data-driven simulation of the supply-chain-insights from the aerospace sector.” *International Journal of Production Economics,* 110(1), 70–84.

Torabi, M., and Mahlooji, H. (2017). “An integrated simulation-DEA approach to multi-criteria ranking of scenarios for execution of operations in a construction project.” *Iranian Journal of Management Studies*, 9(4), 801–827.

U.S. Census Bureau. (2017). “Construction Spending.” (https://www.census.gov/construction/c30/c30index.html) (Dec. 25, 2017).

U.S. Department of Commerce. (2014). “Productivity Growth in Construction.” (http://www.bls.gov/osmr/pdf/ec140090.pdf) (Nov. 28, 2016).

US DHHS, and NIH. (2006). “Your guide to physical activity and your heart.” *NIH Publication*, (06–5714).

Vasenev, A., Hartmann, T., and Dorée, A. G. (2014). “A distributed data collection and management framework for tracking construction operations.” *Advanced Engineering Informatics,* 28(2), 127–137.

Werndl, C. (2009). “What are the new implications of chaos for unpredictability?” *The British Journal for the Philosophy of Science*, 60(1), 195–220.

Wilson, C., Harvey, A., and Thompson, J. (1999). “ClustalG: Software for analysis of activities and sequential events.”

Wilson, W. C. (1998). “Activity pattern analysis by means of sequence-alignment methods.” *Environment and Planning A*, 30(6), 1017–1038.

Wisconsin DHHS. (2017). *Chart of calories burned per hour*.

World Economic Forum. (2016). “Shaping the Future of Construction: A Breakthrough in Mindset and Technology.” (http://www3.weforum.org/docs/WEF_Shaping_the_Future_of_Construction_full_report__.pdf) (Dec. 31, 2017).

Yang, H. (2013). “Solving problems of imperfect data streams by incremental decision trees.” *Journal of Emerging Technologies in Web Intelligence*, 5(3), 322–331.

Yang, K., Ahn, C. R., Vuran, M. C., and Aria, S. S. (2016). “Semi-supervised near-miss fall detection for ironworkers with a wearable inertial measurement unit.” *Automation in Construction*, 68, 194–202.

Yang, K., Aria, S., Ahn, C. R., and Stentz, T. L. (2014). “Automated detection of near-miss fall incidents in iron workers using inertial measurement units.” *Construction Research Congress 2014: Construction in a Global Network*, 935–944.

Ye, J., Coyle, L., McKeever, S., and Dobson, S. (2010). “Dealing with activities with diffuse boundaries.” *Proceedings of Pervasive 2010 workshop on How to do good activity recognition research? Experimental methodologies, evaluation metrics, and reproducibility issue*, Helsinki, Finland.

Yu, J., and Buyya, R. (2006). “Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms.” *Scientific Programming*, 14(3–4), 217–230.

Zamalloa, M. Z., and Krishnamachari, B. (2007). “An analysis of unreliability and asymmetry in low-power wireless links.” *ACM Transactions on Sensor Networks (TOSN)*, 3(2), 7.