Title	Hybrid systems modelling aided by machine learning with applications in healthcare
Author(s)	Elbattah, Mahmoud
Publication Date	2018-03
Publisher	NUI Galway
Item record	http://hdl.handle.net/10379/7260

Hybrid Systems Modelling Aided by Machine Learning with Applications in Healthcare

Mahmoud Elbattah
College of Engineering and Informatics
National University of Ireland Galway

Supervised by

Dr. Owen Molloy
College of Engineering and Informatics
National University of Ireland Galway

This thesis is submitted to the College of Engineering and Informatics, National University of Ireland Galway, in fulfilment of the requirements for the degree of

Doctor of Philosophy

March 2018

Acknowledgements

This work would not have been possible without all who have helped me along the way. First, I would like to dedicate this work to my parents for their unbreakable faith and moral support over long years. Their love and encouragement helped me continue to be persistent and diligent in work.

Second, I would like to express my sincere gratitude to my supervisor Dr. Owen Molloy in many ways. For the support and large degree of freedom I was given to pursue my own research interests. In fact, this has been a key factor in shaping my character as a scholar and researcher. I highly appreciate his valuable advice, attentive supervision, and enthusiasm for my ideas. I am also grateful to his support of my trips to international conferences and summer schools, which opened new frontiers for research collaborations and opportunities.

Furthermore, I would like to acknowledge the truly effective role of the Graduate Research Committee for the evaluation and development of my research. In this regard, many thanks to Dr. Michael Madden, Dr. Michael Schukat, and Dr. Enda Howley for their guidance, advice, criticism, encouragement, and insights throughout my PhD journey.

Equally important, I would like to thank very much the lecturers of modules that I attended at NUI Galway or other Irish institutions. I deeply believe that those modules helped me develop the ideas of my research. In particular, many thanks to:

- Dr. Jerome Sheahan, NUI Galway.
- Ms. Mary Dempsey, NUI Galway.
- Dr. Michael Madden, NUI Galway.
- Dr. Jim Duggan, NUI Galway.
- Dr. Matthias Nickles, NUI Galway.
- Dr. Conor Hayes, NUI Galway.
- Dr. Intizar Ali, NUI Galway.
- Prof. Noel Buckley, University of Limerick.
- Prof. Timothy Jones, Burren College.

Next, I wish to pay tribute to the MOOCs providers including Coursera.org and edX.org. I could not be more fortunate to gain a learning experience from such a supporting and knowledgeable community. Specifically, I am grateful to Prof. Andrew Ng (Stanford University) for his comprehensive course on Machine Learning on Coursera. And special thanks to Prof. Yaser S. Abu-Mostafa (California Institute of Technology) for the wonderful course-Learning from Data. Also, I would like to give special thanks to Dr. Steve Elston (Quantia Analytics) who delivered the 'Data Science and Machine Learning Essentials' course on edX. Dr Steve's tutorials were a valuable source of practical knowledge for developing Machine Learning models on the Azure Machine Learning Studio.

I highly appreciate the generous financial support given by the College of Engineering and Informatics at NUI Galway. The Hardiman Scholarship has been the main source to fund my PhD study over four years.

Last but not least, I convey my sincere thanks to Professor Navonil Mustafee from the University of Exeter in England. Prof. Mustafee kindly served as the external examiner of the dissertation. It must be acknowledged that his insightful comments, suggestions, and thoughts provided much valuable input to the study.

Abstract

Recent trends towards data-driven methods may require a substantial rethinking of the process of developing simulation models. For instance, Machine Learning (ML) has demonstrated great potentials for constructing new knowledge, or improving already established knowledge. Reflecting this trend, the study lends support to the discussion of why and how ML should support the practice of Modelling and Simulation.

Subsequently, the study proposes a hybrid approach towards integrating simulation models with ML. At its core, the approach is based on the premise that system knowledge can be captured and learned in an automated manner aided by ML models. The key idea is to realise ML-guided simulations during different stages of model development.

The study goes through a use case in relation to healthcare, which aims to provide a practical perspective for integrating simulation models with data-driven insights learned by ML models. Through realistic scenarios, the study utilises unsupervised and supervised ML techniques in order to demonstrate the practicality of the approach. First, unsupervised ML (e.g. clustering and rule mining) was utilised in an attempt to discover underlying structures or patterns within patient records. The knowledge discovered represented data-driven insights used to learn about the actual system. In this manner, the data-driven insights helped shape the structure and behaviour of the simulation model. Second, simulation experiments were conducted with the guidance of ML models trained to make predictions on the system behaviour.

From a practical standpoint, it was also aimed to deliver useful insights in relation to healthcare planning in Ireland, with a particular focus on the hip-fracture care scheme. The insights were provided based on a set of simulation models along with ML predictions. At the population level, simulation models were used to mimic the flow of patients, and the care journey, while ML provided accurate predictions of care outcomes at the patient level.

In a broader sense, the study aimed to make the case that simulation models can be integrated with data-driven knowledge learned by ML. In this respect, the integration of simulations and ML can allow for addressing further complex questions and scenarios of analytics. It is believed that the present work contributes in this direction. Recognizing its current limitations, the study can serve as a kernel towards promoting further integration of the practice of Modelling and Simulation with ML.

Table of Contents

Table of Contents (cont'd)

List of Figures

List of Figures (cont'd)

List of Figures (cont'd)

List of Figures (cont'd)

List of Tables

List of Acronyms

Acronym	Nomenclature
ACO	Accountable Care Organisation
AI	Artificial Intelligence
API	Application Programming Interface
ARFF	Attribute-Relation File Format
CHO	Community Health Organisation
CRAN	Comprehensive R Archive Network
CRISP-DM	Cross-Industry Standard Process for Data Mining
CSO	Central Statistics Office
CSV	Comma-Separated Values
DAE	Differential Algebraic Equations
DES	Discrete Event Simulation
DESMO-J	Discrete-Event Simulation and Modelling in Java
DNN	Deep Neural Networks
FPL	Functional Programming Language
GMM	Gaussian Mixture Model
HDFS	Hadoop Distributed File System
HIPE	Healthcare Pricing Office
HIQA	Health Information and Quality Authorit
HPO	Healthcare Pricing Office
HSE	Health Service Executive
IHFD	Irish Hip Fracture Database
KDD	Knowledge Discovery in Databases
LDA	Linear Discriminant Analysis
LOS	Length of Stay
MAS	Multi-Agent System
ML	Machine Learning
M&S	Modelling and Simulation
NHFD	National Hip Fracture Database
NIMS	National Incident Management System
NOCA	National Office of Clinical Audit
ODE	Ordinary Differential Equations
OR	Operational Research

List of Acronyms (cont'd)

Acronym	Nomenclature
PCA	Principal Component Analysis
PCN	Primary Care Network
PCT	Primary Care Team
PDE	Partial Differential Equations
REST	Representational State Transfer
SD	System Dynamics
SEMMA	Sample, Explore, Modify, Model, and Assess
SOM	Self-Organising Map
SQL	Structured Query Language
SSE	Sum of Squared Error
TTA	Time to Admission
TTS	Time to Surgery
UHI	Universal Health Insurance
WHO	World Health Organisation
WSS	Within Cluster Sum of Squares
XML	eXtensible Markup Language

List of Publications

- Elbattah, M., & Molloy, O. (2018, May). A Conceptual Framework for Integrating Simulation Models with Machine Learning. In Proceedings of the 2018 ACM Conference on SIGSIM Principles of Advanced Discrete Simulation. Rome, Italy. ACM. *(In Press)*
- Elbattah, M., & Molloy, O. (2017, December). Learning about Systems Using Machine Learning: Towards More Data-Driven Feedback Loops. In Proceedings of the 2017 Winter Simulation Conference (WSC). Las Vegas, USA.IEEE.
- Elbattah, M., & Molloy, O. (2017, February). Clustering-Aided Approach for Predicting Patient Outcomes with Application to Elderly Healthcare in Ireland. In Proceedings of the 2017 AAAI Conference, Workshop on Health Intelligence. San Francisco, USA. AAAI Publications.
- Elbattah, M., & Molloy, O. (2017, January). Data-Driven Patient Segmentation Using K-Means Clustering: The Case of Hip Fracture Care in Ireland. In Proceedings of 10th Workshop on Health Informatics and Knowledge Management (HIKM), (p. 60). Geelong, Australia. ACM.
- Elbattah, M., & Molloy, O. (2016, September). Using Machine Learning to Predict Length of Stay and Discharge Destination for Hip-Fracture Patients. In Proceedings of SAI Intelligent Systems Conference (pp. 207-217). London, UK. Springer.
- Elbattah, M., & Molloy, O. (2016, July). The Economic Burden of Hip Fractures among Elderly Patients in Ireland: A Combined Perspective of System Dynamics and Machine Learning. In Proceedings of the 34th Conference of the System Dynamics Society. Delft, Netherlands. System Dynamics Society.
- Elbattah, M., & Molloy, O. (2016, July). Bringing Data-Driven Intelligence to Simulation Models Using Machine Learning. Sixth European Business Intelligence & Big Data Summer School (eBISS 2016). Tours, France.
- Elbattah, M., & Molloy, O. (2016, May). Coupling Simulation with Machine Learning: A Hybrid Approach for Elderly Discharge Planning. In Proceedings of the 2016 ACM Conference on SIGSIM Principles of Advanced Discrete Simulation (pp. 47-56). Banff, Canada. ACM.

List of Publications (cont'd)

- Elbattah, M., & Molloy, O. (2015, July). Towards Improving Modeling and Simulation of Clinical Pathways: Lessons Learned and Future Insights. In Proceedings of the 5th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH) (pp. 508-514). Colmar, France.
- Elbattah, M., & Molloy, O. (2018). Analytics Using Machine Learning-Guided Simulations with Application to Healthcare Scenarios. In S. Hawamdeh & H. Chang (Eds.). London, UK. Analytics and Knowledge Management. Taylor & Francis LTD.

Under Review:

- Elbattah, M., & Molloy, O. Mining the Irish Hip Fracture Database: Learning Factors Contributing to Care Outcomes. Submitted to Health Informatics Journal. Sage.

Relevant Collaborations

- Zeigler, B. P., Carter, E. L., Molloy, O., & Elbattah, M. Using Simulation Modeling to Design Value-Based Healthcare Systems. (2016, September). In Proceedings of the Operational Research Society Annual Conference (OR58). Portsmouth, UK. The OR Society.
- Vera-Baquero, A., Colomo-Palacios, R., Molloy, O., & Elbattah, M. (2015). Business Process Improvement by Means of Big Data Based Decision Support Systems: A Case Study on Call Centers. International Journal of Information Systems and Project Management, Vol. 3, No.1.

Chapter 1

Introduction

1.1 Motivation: Integrating Simulation Modelling and Machine Learning

In an early insightful suggestion, Shannon (1975) envisioned that:

> *"The progress being made in Artificial Intelligence technology opens the door for a rethinking of the simulation modeling process for design and decision support."*

The fields of Simulation Modelling and Machine Learning (ML) are long-established in the world of computing. However, they tended to be employed in separate territories with limited integration, if any. This might be explained as follows.

First, the development of ML models is highly data-driven compared to simulation models. Simulation models are largely developed with the aid of domain experts. The subjective perspective of expert knowledge can stipulate to a great extent the behaviour of a simulation model in terms of structure, assumptions, and parameters. While ML models can be constructed with minimal, or even without, involvement of subject matter experts.

From the perspective of data analytics, Simulation Modelling and ML have been often considered to address different types of questions. On one hand, ML is largely concerned with predicting what is likely to happen? On the other hand, simulation models go beyond that question and address further questions of more complex scenarios (e.g. what if?, or how to?). Figure 1.1 illustrates the position of Simulation Modelling and ML within the spectrum of data analytics. The figure classifies analytics into four categories of analytical sophistication.

The initial motivation for this study was to investigate how simulation models can be further integrated with ML. The main view was that ML can be utilised as an assistive tool to support the practice of Modelling and Simulation (M&S) at different stages of model development. For instance, Simulation Modelling can avail of ML for automating or semi-automating the process of capturing system knowledge. The next chapters present further perspectives on the benefits of that integration.

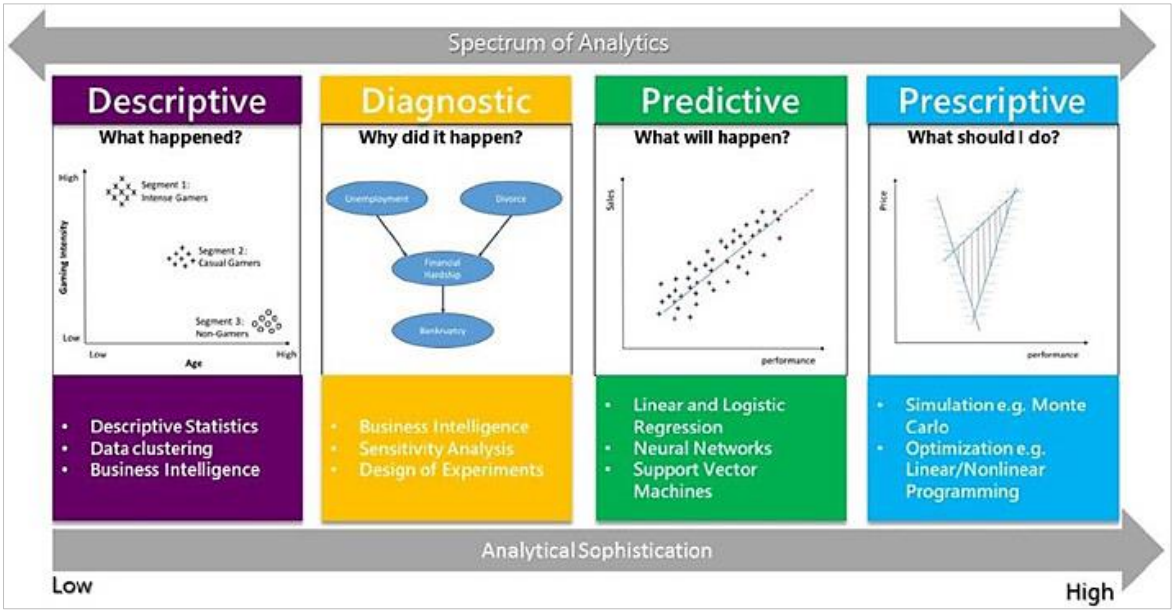


Figure 1.1: The spectrum of data analytics (Barga, Fontama, Tok, & Cabrera-Cordon, 2015).

1.2 Research Questions

How can Machine Learning Assist Learning about Systems?

In an insightful analysis, Sterman (1994) asserted that:

> *"The challenge is how to move from generalisations about accelerating learning and systems thinking to tools and processes that help us understand complexity, and design better policies."*

Sterman was notably alluding to the importance of developing assistive tools that can support the process of learning about systems, and understanding their complexity. The complexity of systems can be interpreted in terms of several dimensions. One possible dimension can be attributed to the data and metadata that represent the system knowledge. Systems can now be dealing with extraordinary amounts of data (i.e. Big Data). Due to the ubiquity of Big Data, system knowledge can accordingly become more contingent on empirical data accumulated or generated rapidly.

In this regard, it should be taken into account that systems involved within such Big Data scenarios place further burdens on the modelling process, which can go beyond human capabilities in many aspects. For instance, it may be required to elicit knowledge from high-dimensional data silos, or dynamic data streams being accumulated with a high velocity. Therefore, the complexity of systems can also be interpreted in terms of the complexity of data encompassing the system knowledge.

Given the significant momentum gained by data analytics over the past years, data-driven knowledge can be utilised to understand, describe, and learn about systems to a greater extent. In view of that, the key impetus for the study was that data-driven methods (e.g. ML) should be further considered as a valid path for the task of learning about systems.

To focus the study, a set of motivational questions were specifically addressed in this regard (see Table 1). Throughout the thesis chapters, these questions are discussed with practical examples that can demonstrate how ML can support the practice of Modelling and Simulation.

Table 1.1: Motivational questions.

Question	Motivation
Q1) How can ML be employed to assist the conceptualisation of systems?	Utilising ML as an assistive tool within the process of learning about the structure or behaviour of systems.
Q2) Is it possible to integrate mental models with ML models in a way that supports the learning process to be developed based on a more data-driven manner? If so, how?	The limitations of our mental models raise a need to consider more relatively unbiased reasoning methods for exploring and describing systems, and designing the corresponding models.
Q3) Which ML techniques can be appropriate for the perception of the structure, or behaviour of systems involved within a problem?	Exploring the possible approaches/methods (e.g. supervised or unsupervised learning) to avail of ML for the purpose of learning about systems.
Q4) Can the integration of ML lead to a higher level of confidence in simulation models, indicated by the accuracy of ML models?	The predictive accuracy of ML models can be more measurable. This may in turn extend the confidence in simulation models designed based on insights from ML models in tandem with mental models.

Modeling Dynamic Behaviour of Systems Aided by Machine Learning

Jay Forrester (1994) described a model as a theory of behaviour, which represents the way in which some parts of the real system works. Based on the system knowledge, a simulation model is developed to model such behaviour with some level of realism.

However, real-world systems inherently exhibit adaptive, dynamic, goal-seeking, self-preserving, and sometimes evolutionary behaviour, making a system more than simply the sum of its parts (Meadows, 2008). Modelling and Simulation is the discipline that can harness that inherent complexity of systems, and gain insights into how real-world systems actually work. However, a simulation model may become an inadequate representation of reality over time with such dynamic behaviour of systems. This is what can be described as the 'Concept Drift' (Schlimmer & Granger, 1986). The Concept Drift term refers to modelling a non-stationary problem over time, where a changing context can lead to a mismatch between models and actual problems.

In this respect, the study considered the following question: *what if a simulation model can become aware of new information or knowledge in an automated manner?* In other words, what if a simulation model can adapt to new situations without being explicitly informed by a modeller or simulationist? In fact, that hypothetical question represented a key motivation for the study. The main effort was directed towards investigating the possibilities of developing hybrid simulation models that are aided by ML. The questions below will be discussed in the coming chapters.

1. When to consider that a simulation model has 'learned'?
2. How simulation models can learn about changes in the system's behaviour or conditions with a minimal involvement of modellers or simulationists?
3. Is it possible to integrate simulation models with ML models in a way that can enable that learning process to happen in an automated manner? If so, how?

1.3 Study Hypothesis

In light of the mentioned motivations and questions, the primary hypothesis of the thesis was formulated as follows:

The structure or behaviour of simulation models can be designed, adjusted or adapted in accordance with reality via data-driven knowledge learned by Machine Learning models, which are incrementally trained to represent the behaviour of the system of interest.

1.4 Contributions

The contributions of the thesis can be divided into two main parts as follows. The first part of contributions relates to the practice of Simulation Modelling from a methodological perspective. In this regard thesis introduced a conceptual framework that can serve as a guide to help develop the integration of simulation models and ML in a consolidated manner. In this respect, we discussed how ML models can be effectively coupled with simulation models in order to realise adaptive computer simulations that can learn to change their behaviour in accordance with changes in system's behaviour. In conjunction with simulation experiments, the trained predictors are used to guide the simulation model by making predictions on key variables that represent behavioural changes.

Further, the thesis demonstrated how ML can be effectively used as an assistive tool during the process of problem conceptualisation and learning about systems. Specifically, the study applied knowledge learned from unsupervised ML in order to design simulations that effectively represent actual systems in terms of structure or behaviour. For the purpose of demonstration, System Dynamics (SD) models were designed predicated on data-driven insights learned by ML clustering and rule mining experiments.

The second part of contributions is considered from a practical standpoint pertaining to the healthcare use case adopted. In this regard, the study is claimed to deliver useful data-driven insights in relation to the expected demand for healthcare due to population ageing, with a particular focus on hip fracture care in Ireland. The output of the simulation model along with ML predictions realised a population-based perspective of the demand for hip fracture care, regarding elderly patients in particular. The incorporation of ML is claimed to improve the

predictive power of the simulation model, in turn improving its validity and credibility for decision making.

The results provided insights at two integrated levels for healthcare planning. At the population level, simulation models were used to mimic the flow of patients and the care journey, while ML provided accurate predictions of care outcomes at the patient level. The insights were provided based on a well-rounded picture corresponding to the demographic profiles, structure, and capacity of the healthcare system in Ireland.

1.5 Thesis Organisation

The structure of the thesis is organised as follows. This chapter introduced the key motivation along with a presentation of the research questions, hypothesis, and contributions.

Chapter 2 laid the foundational background necessary for the rest the thesis chapters. Initially, the chapter provided a brief overview of the main concepts of Modeling and Simulation, ML, and to the analytics tools and technologies used by the study. Furthermore, the background described the healthcare system in Ireland, and the dataset used. Eventually, the chapter discussed related studies from different contexts including: Hybrid Simulations, ii) AI-Assisted Simulations, iii) Simulation-Based Healthcare Planning, and iv) Applications of Machine Learning in Healthcare.

Chapter 3 presents a conceptual framework to guide the implementation of integrating simulation models with ML. The presented approach is based on the premise that system knowledge can be captured and learned in an automated manner aided by ML. The approach is conceived to help realise adaptive simulation models that can learn to change their behaviour in response to behavioural changes in the system of interest.

Chapter 4 presents data-driven insights from the healthcare dataset used by the study. The knowledge discovery process included the use of data clustering and rule mining. The knowledge gained will be utilised in the following chapters within building simulation models of the patient's journey.

Chapter 5 starts with a discussion of why and how ML can support the practice of Modelling and Simulation. It is aimed to demonstrate that the integration of mental models with data-driven insights learned by ML models can yield potential benefits for the practice of modelling and simulation. Practical scenarios are presented to illustrate how ML can assist the process of problem conceptualisation. Based on a use case in relation to healthcare, the chapter provides a pragmatic perspective for integrating SD models with data-driven insights learned by ML models.

Chapter 6 go through the development of supervised ML models in order to predict care outcomes for elderly patients who undergo hip fracture care in Ireland. Based on patient records, a regression model and classifier were developed to predict the inpatient LOS, and discharge destination respectively. The developed predictors are claimed to make predictions on those outcomes with high accuracy. Further, the prediction models were deployed as predictive web services provided by the Azure platform.

Chapter 7 presents a hybrid modelling approach that integrated Simulation Modelling with ML including unsupervised and supervised techniques at different stages of model development. A practical scenario was presented in relation to healthcare planning in Ireland to demonstrate the applicability of the approach. First, the knowledge learned by unsupervised ML models was used to assist the modelling phase. Second, simulation experiments were conducted with the guidance of ML models trained to make predictions on the system's behaviour. From a practical standpoint, the chapter also delivered useful insights in relation to the expected demand for hip fracture care due to population ageing. The insights were provided based on a well-rounded picture corresponding to the demographic profiles, structure, and capacity of the healthcare system in Ireland.

Chapter 8 discussed the idea of realising adaptive simulation models aided by ML. Guided by ML predictions, it was demonstrated how a simulation model can adjust its behaviour to reflect on changes in the actual system behaviour. As such, simulation models can learn to change their behaviour in accordance with system's behavioural changes in an automated manner, or with minimal human input. The chapter utilised the healthcare use case, simulation models, and ML models developed previously to validate the approach.

Chapter 9 provides a set of conclusions to thesis. Answers to research questions are provided along with directions for future work.

Chapter 2

Background

2.1 Modelling and Simulation

> *"When experimentation in the real system is infeasible, simulation becomes the main, and perhaps the only, way to discover how complex systems work"*, Sterman (1994).

Simulation Modelling was considered as a standalone discipline that encompasses designing a model of an actual or theoretical system, executing the model on a digital computer, and analysing the execution output (Fishwick, 1995a). Shannon (1975) described simulation as the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system. With a virtual build-and-test environment, simulation provides the feasibility to model real systems that exhibit adaptive, dynamic, goal-seeking, self-preserving, and sometimes evolutionary behaviour (Meadows, 2008).

Newell and Simon (1959) asserted that the real power of the simulation approach is that it provides not only a means for stating a theory, but also a very sharp criterion for testing whether the statement is adequate. Similarly, Forrester (1968) emphasised in one of his principles of systems that simulation-based solutions present as the only feasible approach to represent the inter-dependence and non-linearity of complex systems, whereas analytical solutions can be largely impossible.

Figure 2.1 provides an illustration of the process of constructing simulation models from conceptual modeling to implementation. The figure distinguishes the levels of abstraction at each stage.

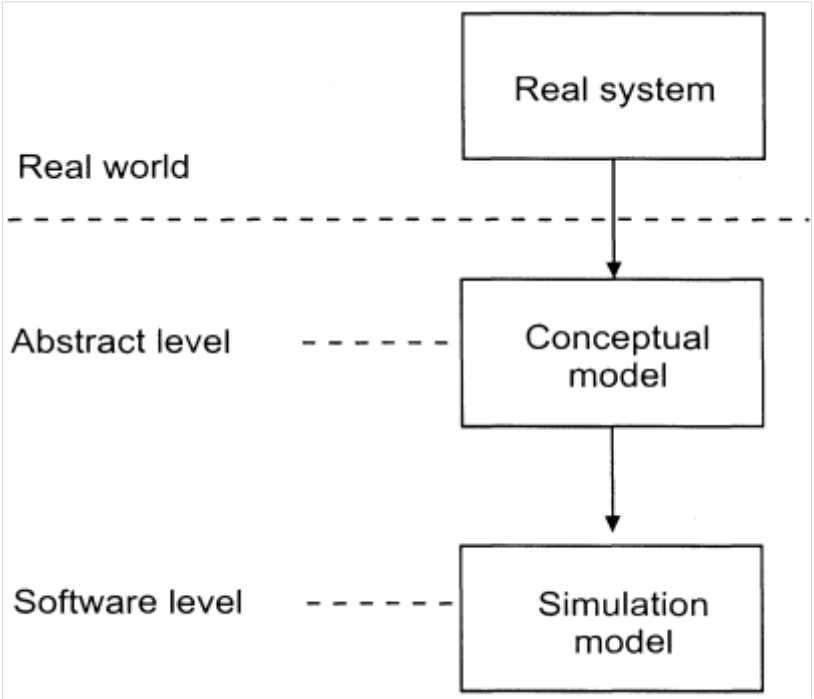


Figure 2.1: The development of simulation models (Garrido, 2012).

A simulation model is developed largely based on the 'world view' of a modeller. A world view reflects how a real-world system is mapped to a simulation model. In this respect, there are three primary approaches including: i) System Dynamics (SD), Discrete-Event Simulation (DES), and iii) Agent-Based modelling.

The SD approach assumes a very high degree of abstraction, which can be considered adequately for strategic modelling. On the other hand, discrete-event models maintain medium and medium-low abstraction, whereas a model comprises a set of individual entities that have particular characteristics in common. Agent-based models are positioned in an intermediate position, which can vary from very fine-grained agents to the highly abstract models. Figure 2.2 portrays the three approaches with respect to the level of abstraction. Further, Table 2.1 makes a more detailed comparison based on (Brailsford, & Hilton, 2001; Lane, 2000; Sweetser, 1999).

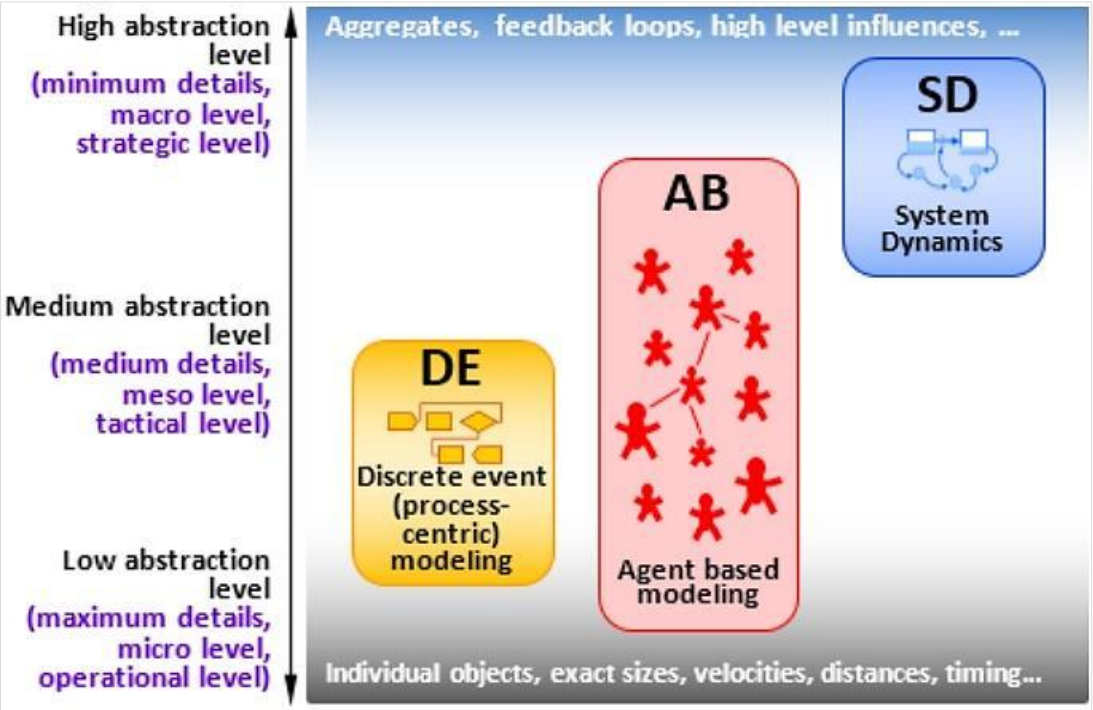


Figure 2.2: Simulation modelling approaches (Borshchev, 2013).

Table 2.1: Comparison between the SD and DES approaches.

	System Dynamics Models	**Discrete-Event Models**
Scope	Operational / Tactical	Strategic
Nature of Models	Continuous / Mostly deterministic	Discrete / Mostly Stochastic
World View	A series of stocks and flows, in which the state changes are continuous. Entities are viewed as a continuous quantity, or aggregates rather than individual entities (e.g. products, people).	Entities flowing through a network of queues and activities. Entities have attributes that describe specific features of an entity (e.g. entity type, dimensions, weight, priority).
Building Blocks	Stocks, in-flows, outflows, tangible variables, and soft variables.	Individual entities, attributes, queues and events.
Model Outputs	A full picture of the system.	Estimates of system's performance.

Conceptual Modelling

Conceptual modelling has been recognised as one of the most difficult issues within simulation studies. Robinson (2015) defined conceptual modeling as the abstraction of a simulation model from the part of the real world it represents (i.e. actual system). A more formal definition, a conceptual model is a non-software specific description of the computer simulation model, describing the objectives, inputs, outputs, content, assumptions and simplifications of the model (Robinson, 2008).

With the knowledge of objectives, inputs and outputs, the model content can be informed. According to (Robinson, 2015), the model content contains two dimensions as follows:

- **The scope of the model:** The model boundary or the breadth of the actual system that is to be included in the model.
- **The level of detail:** The details to be included for each component in the model's scope.

Artefacts of Conceptual Modelling

Robinson (2011) articulated a further understanding of the conceptual modelling process from a wider perspective as in shown in Figure 2.3. The figure illustrates the key artefacts of conceptual modelling. The 'cloud' represents the real-world environment where the problem under study exists. The four rectangles represent specific artefacts of the conceptual modelling process as below:

- **System Description:** A description of the problem, and real-world elements related to the problem.
- **Conceptual model:** As defined above.
- **Model Design:** The design of the constructs for the computer model (e.g. data, components, model execution, etc.) (Fishwick, 1995b).
- **Computer Model:** A software-specific representation of the conceptual model.

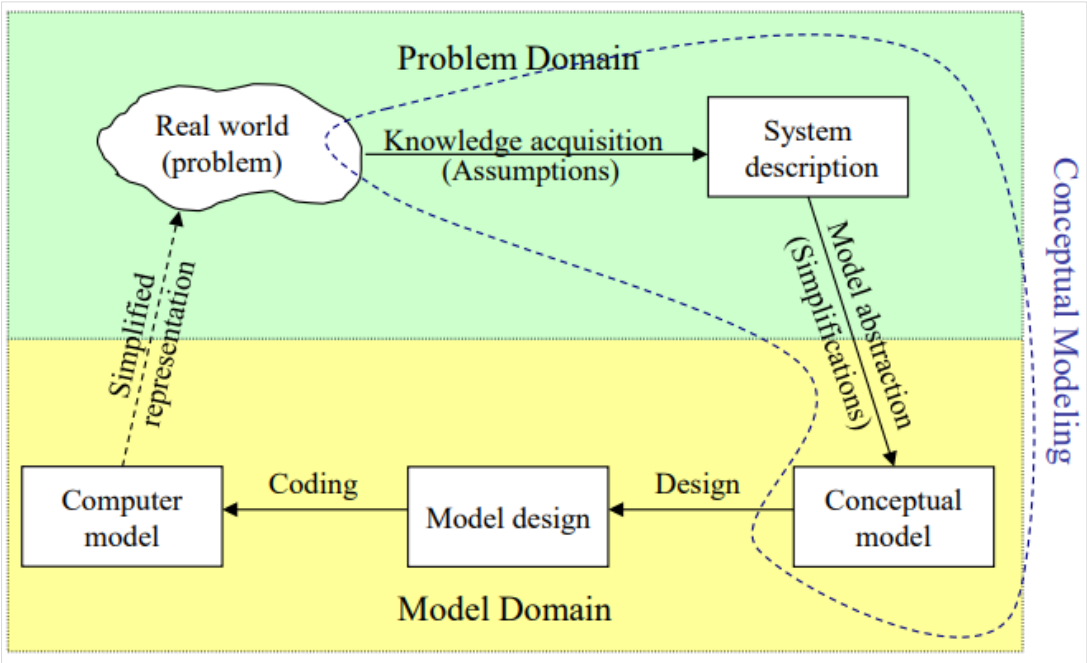


Figure 2.3: The artefacts of conceptual modelling (Robinson, 2011).

Frameworks for Conceptual Modelling

A framework for conceptual modelling provides a set of steps and tools that guide a modeller through the development of a conceptual model. It is also useful for teaching the practice of conceptual modelling. Examples included in the simulation literature are:

- Conceptual modelling framework for manufacturing (van der Zee, 2007).
- The ABCmod conceptual modelling framework (Arbez, & Birta, 2011).
- Conceptual modelling with Onto-UML (Guizzardi, & Wagner, 2012).
- Conceptual Model Development Tool (KAMA), Federation Development and Execution Process (FEDEP), Conceptual Models of the Mission Space (CMMS), Defense Conceptual Modeling Framework (DCMF), and Base Object Model (BOM), (Karagöz, & Demirörs, 2011).
- Conceptual modelling using the Structured Analysis and Design Technique (SADT) (Ahmed, Robinson, & Tako, 2014).
- The PartiSim framework (Tako, & Kotiadis, 2015).

Sources of System Knowledge

By and large, a simulation model attempts to formulate a representation of the actual system in order to answer questions about that system. As described by (Cellier, 1991), a model for a system (S) and an experiment (E) is anything to which (E) can be applied in order to answer questions about (S). In other words, every simulation model attempts to formulate a representation of system knowledge in order to answer questions about that system. This section reviews the common sources of knowledge that can be utilised to develop simulation models, and the potential links to the context of Big Data as well.

System knowledge can be learned from different sources. Huang (2013) mentioned four categories of knowledge that can be utilised to build simulation models as below:

- **Formal Knowledge:** This kind of knowledge can be explicitly presented and maintained. Examples of formal system knowledge can be theories, theorems, or mathematical models. The content of formal knowledge has the advantage of being readily accessible in its meaning and form.
- **Informal Knowledge:** On the other hand, the informal knowledge is implicitly represented in an intangible format in the heads of people who are experienced in the system, (e.g. domain experts, or system users). Therefore, this kind of knowledge needs to be formulated using elicitation and formalisation methods (e.g. meetings, interviews, or questionnaires).
- **Empirical Data:** Refers to the type of data obtained from recorded observations or measurements about the system. It potentially contains information about the behaviour of a system and its sub-systems.
- **System Description:** Refers to data that can contain descriptive information about the system in terms of structure and relation information. Compared to empirical data, system description characterises a system and its sub-systems themselves. This type of knowledge is often produced by people such as domain experts and or engineers involved with designing the system. Examples are documents, floor-plans of factories, manufacturing process maps. Figure 2.4 sketches the sources of system knowledge.

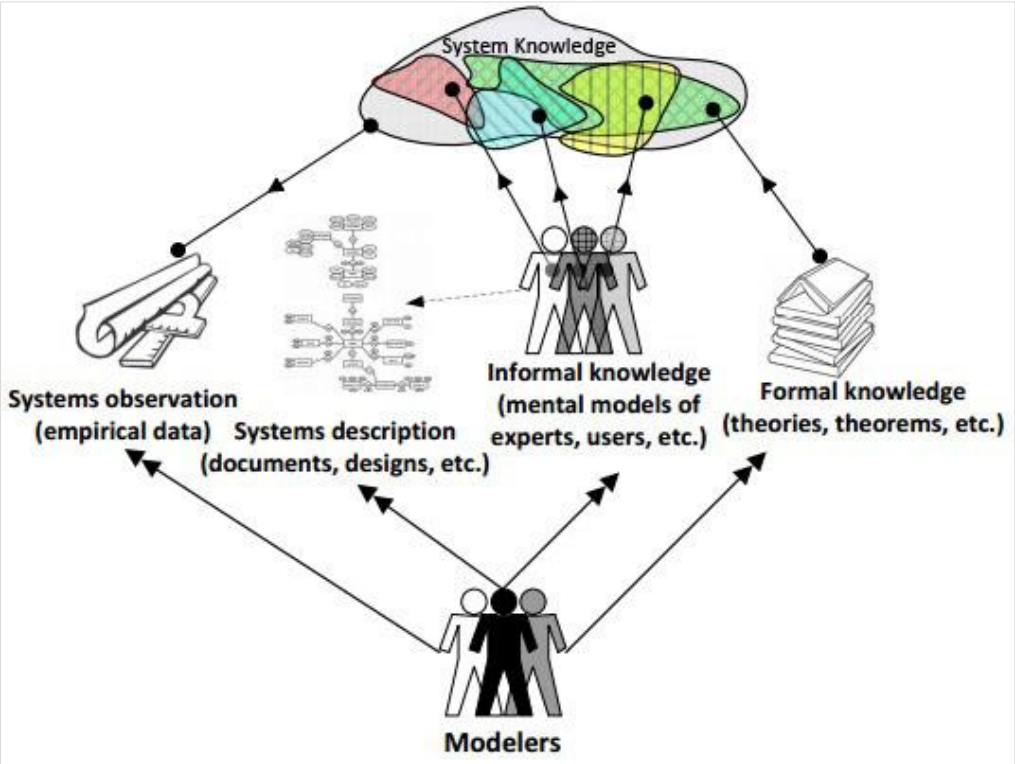


Figure 2.4: Sources of system knowledge (Huang, 2013).

In the era of Big Data, it should be considered that system knowledge will increasingly rely on empirical data accumulated or generated autonomously. Specifically, more data will be increasingly utilised to learn about systems. Therefore, systems dealing with Big Data scenarios will inevitably place further burdens on the modelling process, which can go beyond the capabilities of humans in many respects. For instance, the system knowledge can exist underlying huge amounts of data, or can be continuously accumulated with high velocity.

In this regard, the community of M&S has started to consider the opportunities and challenges facing the development of simulation models in an age marked by data-driven knowledge. For instance, (Taylor et al., 2013) introduced the term *Big Simulation* to describe one of the grand challenges for the development of simulation models. Big Simulation addressed issues of scale for Big Data input, very large sets of coupled simulation models, and the analysis of Big Data output from these simulations, all running on a highly distributed computing platform. Other insightful studies (e.g. Tolk, 2015; Tolk et al., 2015) emphasised the need for integrating the practice of M&S with Big Data. In particular, it was stressed that Big Data techniques and technologies should be considered in order to avail of rapidly accumulating data that may be unstructured as well.

2.2 Machine Learning

The field of ML has come into prominence as one of the cornerstones of the area of Artificial Intelligence (AI). In essence, ML focuses on algorithms that can learn from data. The power of learning from data is that the entire process can be automated with a slight need for prior domain knowledge (if any). This is a very attractive approach that has spread rapidly beyond the discipline of Computer Science. ML techniques are now heavily used in a variety of fields including search engines, recommender systems, marketing personalisation, medical assistive tools, automatic translation, financial trading, and many other applications. Moreover, ML algorithms have gained further popularity due to their potential to process large and high-dimensional datasets. Figure 2.5 and Figure 2.6 sketch the main difference between traditional programming and ML as highlighted above.

The outcome of the learning process can be two-fold. On one hand, ML can provide descriptive knowledge in terms of understanding the inherent structure and properties of data (i.e. unsupervised learning). On the other hand, predictive capabilities can enable to make predictions of future data examples (i.e. supervised learning).

It is noteworthy that the field of ML has a broad diversity of theoretical and practical tracks. A detailed presentation of those aspects would go beyond the scope and of this study and space available. Therefore, this section merely aims to provide an overview of ML including its definitions, approaches, and practical considerations.

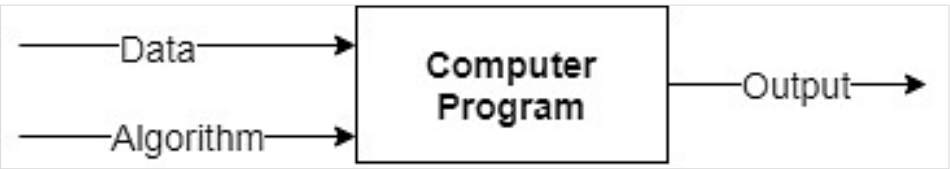


Figure 2.5: Traditional programming.

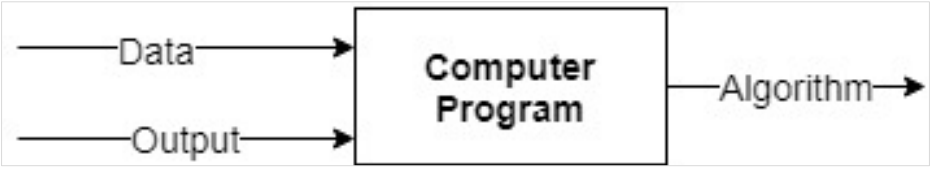


Figure 2.6: Machine Learning.

Definitions of ML

ML builds on concepts from a diversity of fields and disciplines including statistics, AI, information theory, biology, philosophy, cognitive science, computational complexity, and others. Tom Mitchell, one of the pioneers in ML, recommended that the field of ML should be viewed from all of those perspectives (Mitchell, 1997). This section reviews distinct interpretations of ML, and learning in a broader sense as below:

- A broad definition of learning is given by the Oxford Dictionary (Stevenson, 2010) as to become aware of something by information or from observation.
- Samuel (1959), one of the early founders of that field, defined ML as the subfield of computer science that gives computers the ability to learn without being explicitly programmed.
- Simon (1983) described the process of learning as changes in the system that are adaptive in the sense that they enable the system to do the same task(s) more efficiently and more effectively the next time.
- Ian Witten similarly interpreted leaning as that things learn when they change their behaviour in a way that makes them perform better in the future (Witten, Frank, & Hall, 2005).
- A more formal definition was formulated by Tom Mitchell as follows. A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E (Mitchell, 1997). For example, consider an example of a program that attempts to learn how to play the checkers game:

  (T) → Playing checkers,

  (P)→ Percent of games won against opponents, and

  (E) → Playing practice games against itself.

Components of Machine Learning

According to (Domingos, 2012), the process of building ML models should include three main components as follows:

- **Representation:** An ML model must be represented in a formal language that can be handled by computers. Choosing a representation for a learner is equivalent to choosing the set of classifiers that it can possibly learn. This set is called the hypothesis space of the learner. If a classifier is not in the hypothesis space, it cannot be learned. Examples of representation include sets of rules, instances, decision trees, neural networks, graphical models, support vector machines, and others. A related question is how to represent the input, or which features to use.
- **Evaluation:** The method used to evaluate candidate hypotheses. Different naming can be used to refer to this including the evaluation function, objective function, or scoring function. The evaluation function used internally by the learning algorithm may differ from the target function that the algorithm seeks to optimise. Examples are precision and recall, squared error, information gain, and others.
- **Optimisation:** Finally, a method is needed to search among the models to find the highest-scoring one. The selection of optimisation technique is of significant importance to the model efficiency, and also helps determine the best model if the evaluation function has more than one optimum. Examples are gradient descent optimisation, and linear programming.

Table 2.2 shows common examples of each of the above-mentioned components. For example, the K-Nearest neighbour algorithm classifies a test example by finding the (K) most similar training examples and predicting the majority class among them. Hyperplane-based methods form a linear combination of the features per class and predict the class with the highest-valued combination. Decision trees test one feature at each internal node, with one branch for each feature value, and have class predictions at the leaves.

Table 2.2: The key components of ML with examples.

Representation	Evaluation	Optimisation
Instances K-Nearest Neighbour Support Vector Machines Decision Trees Sets of rules Propositional Rules Logic Programs Neural Networks Model Ensembles Graphical models Bayesian networks	Accuracy/Error Rate Precision and Recall Squared Error Information Gain Margin Likelihood Posterior Probability	Combinatorial Optimisation Greedy Search Beam Search Branch-and-Bound Unconstrained Continuous Optim. Gradient Descent Conjugate Gradient Quasi-Newton Methods Constrained Continuous Optim. Linear Programming Quadratic Programming

Supervised Learning

Supervised ML algorithms are trained using labelled examples, such as an input where the desired output is known (i.e. Y = f (X)). The learning algorithm receives a set of inputs along with the corresponding correct labels, and the algorithm learns by comparing its predicted labels to the actual ones to find errors. The model can then be modified to minimise that error. Supervised learning is commonly used in applications where historical data predict likely future events. For example, you can anticipate when credit card transactions are likely fraudulent, or that the insurance client can file a claim.

Supervised learning problems can be broadly categorised in two types:

- **Regression:** To predict the outcome of a given sample where the output variable is in the form of real values. Examples include real-valued labels such as temperature, amount of rainfall, etc.
- **Classification**: To predict the outcome of a given sample where the output variable is in the form of categories or classes. Examples include labels such as classifying cancer patients into high or low risk groups.

In the supervised learning, the ultimate goal is to develop hypothesis that best generalises the data samples. The process of learning happens in a form of optimisation to find a hypothesis that can accurately predict some interesting value (i.e. h(x)). Figure 2.7 illustrates the supervised learning process.

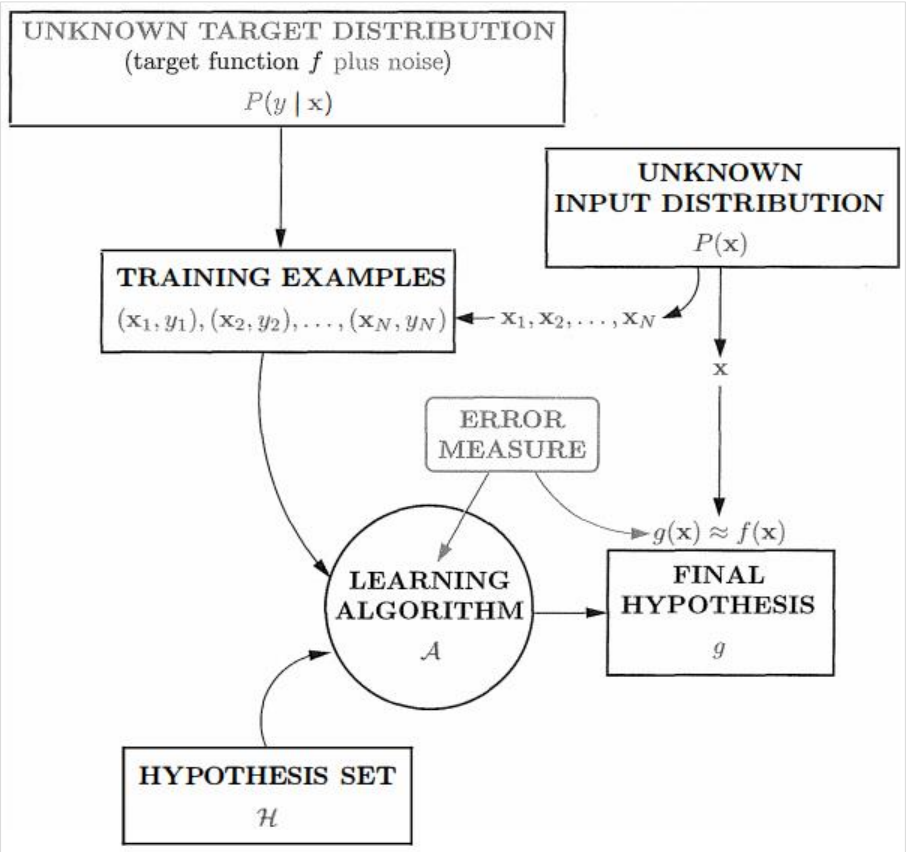


Figure 2.7: The supervised learning process (Abu-Mostafa, Magdon-Ismail, & Lin, H.T, 2012).

Unsupervised Learning

In unsupervised learning problems, the training data do not include any output information (i.e. labels) at all. Unsupervised learning algorithms use unlabelled training data to model the underlying structure of data. For instance, consider the task of categorising patients into coherent groups. Those patients who have similar properties (e.g. age, gender, symptoms) can be put together in one category, without naming that category. Unsupervised learning tasks can be of three types as follows.

Association Rule Mining

To discover the probability of the co-occurrence of items in a collection of transactions. Different association rules express different regularities underlying a dataset, and they therefore predict different aspects. The Apriori algorithm is one of the popular algorithm used for rule mining (Agrawal, Imieliński, & Swami, 1993; Agrawal & Srikant, 1994). It has been extensively used in applications for market-basket analysis.

It is noteworthy that association rules are different from classification rules in two respects as follows:

- Association rules can predict any attribute, not only the class, and this gives them the freedom to predict the co-occurrence or combinations of attributes.
- In contrast to classification rules, association rules are not intended to be used together as a set.

A number of measures can be used to evaluate the interestingness and usefulness of discovered rules. The coverage of an association rule is the number of instances for which it predicts correctly, named as *Support*. The accuracy of rules is called *Confidence*, expressed as a proportion of all instances to which it applies. Another measure named as *Lift*, which gives an indication of rule significance. Specifically, lift represents the predictive advantage a rule offers over simply guessing based on the frequency of the rule consequence (Halkidi, & Vazirgiannis, 2009). Thus, lift may be an indication whether a rule could be considered as representative of the data, and be used in the process of decision-making (Bayardo, Agrawal, & Gunopulos, 1999). Figure 2.8 summarises those three measures along with their equations.

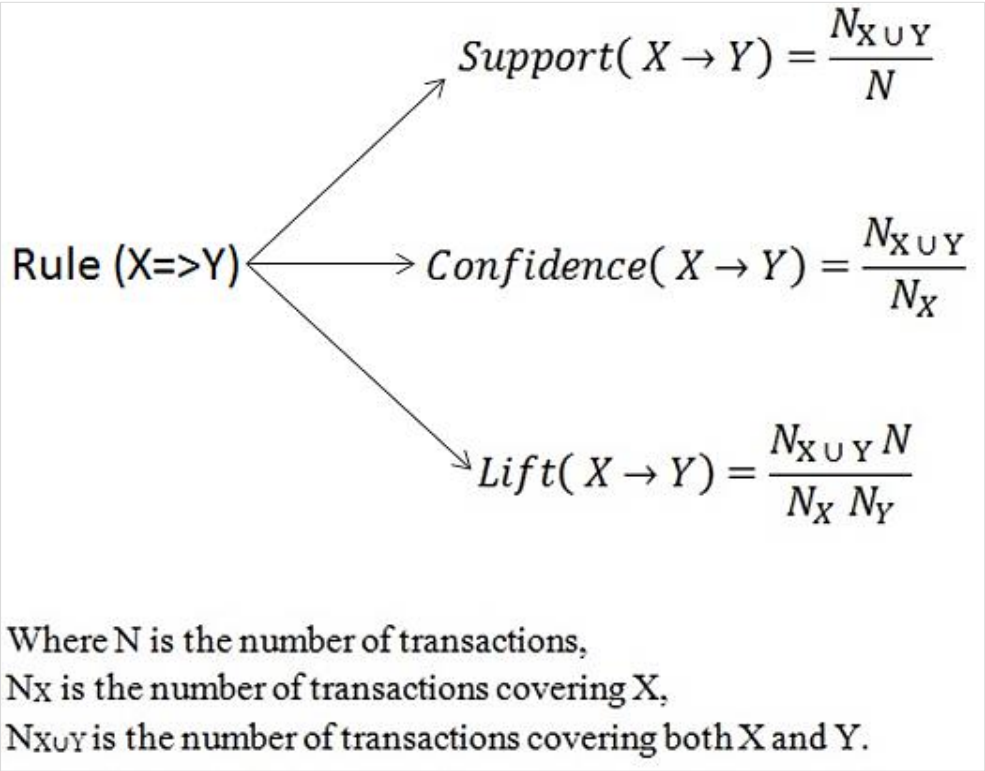


Figure 2.8: Measures used to evaluate association rules.

Clustering

A broad definition of clustering was given by (Aldenderfer, & Blashfield, 1984) as the segmentation of a heterogeneous population into a number of more homogeneous subgroups. Clustering is an effective unsupervised technique for analysing large datasets without making any prior assumptions. Identifying clusters in data can be useful in helping to identify particular points in the data that show similar feature values to each other.

Different tasks can be served by clustering including: i) Exploring the underlying structure of data, ii) Discovering meaningful patterns, and iii) Summarising key characteristics of data. With such tasks, data clustering was adopted in a variety of applications for a wide range of domains ranging from statistics, computer science, and biology to social sciences or psychology.

A variety of algorithms has been developed for data clustering purposes. It was argued that the reason behind that abundant number of clustering methods is due to that the notion of cluster is not precisely defined (Estivill-Castro, & Yang, 2000). Accordingly, many clustering methods have been developed based on a different induction of clusters. While an exhaustive discussion of clustering algorithms may not possible here, this section overviews the two main families of clustering approaches: i) Hierarchical Clustering Methods, ii) Partitioning Clustering Methods. The rest of this section discusses the concepts and mathematics underlying those approaches. The discussion was mainly based on the review given by (Rokach, 2009).

Hierarchical Clustering Methods

This family of clustering methods attempts to construct clusters by recursively partitioning data instances in either a top-down or bottom-up fashion. These methods can be sub-divided as following:

- **Agglomerative Clustering**: Each data point initially represents a cluster of its own. Then clusters are successively merged until the desired cluster structure is realised.
- **Divisive Clustering:** All data points initially belong to one cluster. Then that cluster is divided into sub-clusters, which are successively divided into their own sub-clusters. This process continues until the desired cluster structure is obtained.

The result of the hierarchical methods is a dendrogram (see Figure 2.9), representing the nested grouping of objects and similarity levels. Clusters can be obtained by cutting the dendrogram at the desired similarity level.

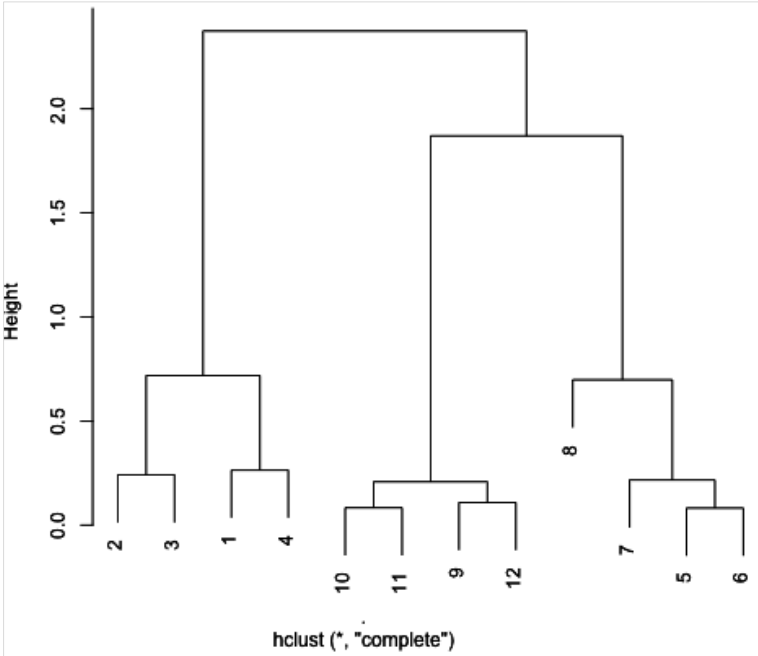


Figure 2.9: Example of a cluster dendrogram (Roger, 2015).

Partitioning Clustering Methods

Partitioning methods relocate data points by moving them from one cluster to another, starting from an initial, usually random, partitioning. In contrast to hierarchical methods, partitioning methods require that the number of clusters to be pre-set by the user. To achieve global optimality in partitioned-based clustering, an exhaustive enumeration process of all possible partitions is required. Because this is not feasible, certain greedy heuristics are used in the form of iterative optimisation. Namely, a relocation method iteratively relocates points between the K clusters. The K-Means algorithm is a typical example of this category of algorithms.

As the name suggests, the K-Means algorithm partitions data into K clusters represented by their centroids (i.e. centre or mean). The centroid of each cluster is calculated as the mean of all the instances belonging to that cluster. The basic idea is to find a clustering structure that minimises a certain error criterion that measures the distance (e.g. Euclidian distance) of each instance to its representative value.

In K-Means, the Sum of Squared Error (SSE) is used to measure the total squared distance of points to their representative values. SSE may be globally optimised by exhaustively enumerating all partitions, or by giving an approximate solution based on heuristics.

Figure 2.10 presents the pseudo-code of the K-Means algorithm. The algorithm starts with an initial set of cluster centres, chosen at random or according to some heuristic procedure. In each iteration, each instance is assigned to its nearest cluster centre according to the Euclidean distance between the two. Then the cluster centres are re-calculated. The centre of each cluster is calculated as the mean of all instances belonging to that cluster:

$$\mu_k = \frac{1}{N_k}\sum_{q=1}^{N_k} x_q$$

Where N_k is the number of instances belonging to $cluster_k$ and μ_k is the mean of the cluster.

```
Input:  S (instance set), K (number of cluster)
Output:  clusters
 1:  Initialize K cluster centers.
 2:  while termination condition is not satisfied do
 3:      Assign instances to the closest cluster center.
 4:      Update cluster centers based on the assignment.
 5:  end while
```

Figure 2.10: K-Means pseudocode.

A number of convergence conditions are possible. For example, the search may stop when the partitioning error is not reduced by the relocation of the centres. This indicates that the present partition is locally optimal. Other stopping criteria can be used also such as exceeding a pre-defined number of iterations.

Dimensionality Reduction

Unsupervised ML algorithms generally seek for meaningful relationships or patterns in raw data. The high dimensionality (i.e. number of attributes, or groups of attributes) constitutes a significant hurdle to realise that goal. Higher dimensionality of the input increases the size of the search space in an exponential manner. For instance, it is well-known that the required number of labelled samples for supervised classification increases as a function of dimensionality (Jimenez, & Landgrebe, 1998). Fukunaga (1990) showed that the required number of training samples is linearly related to the dimensionality for a linear classifier and to the square of the dimensionality for a quadratic classifier. It has been estimated that as the number of dimensions increases, the sample size needs to increase exponentially in order to have an effective estimate of multivariate densities (Hwang, Lay, & Lippma, 1994). This phenomenon is usually called the 'curse of dimensionality' (Bellman, 1961).

Four main arguments were recognised for the need of dimensionality reduction (Chizi & Maimon, 2009). In fact, each argument can be referred to as a distinctive sub-problem:

- Decreasing the learning cost.
- Improving the model performance.
- Reducing irrelevant dimensions.
- Reducing redundant dimensions.

Viewed this way, it can be understood that dimensionality reduction may be divided into two main sub-problems as follows:

Feature Selection: The process of feature selection is necessary in order to identify a set of significant features in the dataset, and accordingly discard irrelevant or redundant features. The efficient selection of features reduces the dimensionality of data, and enables learning algorithms to operate faster and more effectively.

Record Selection: Similarly to features, some records (i.e. data examples) may be more useful for the learning process in order to improve the generalisation error (Blum, & Langley, 1997). Figure 2.11 summarises the different aspects of the dimensionality reduction problem.

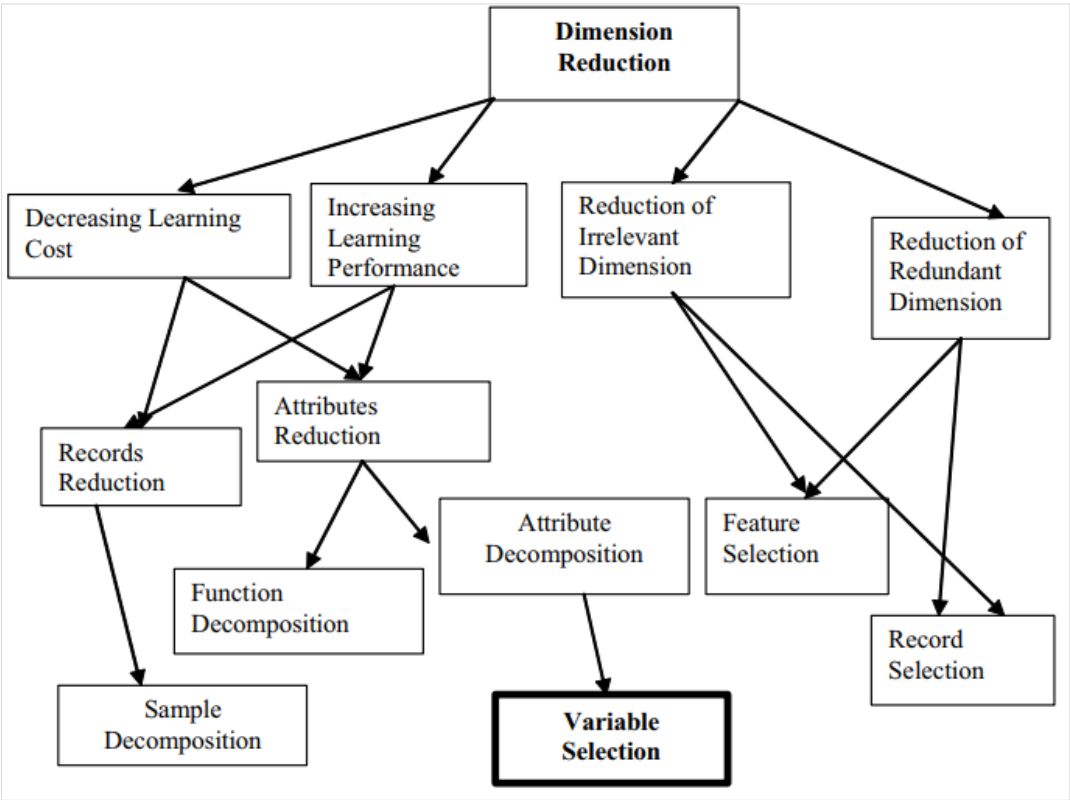


Figure 2.11: Taxonomy of dimensionality reduction problem (Chizi, & Maimon, 2009).

Variable Selection

Different approaches have been developed for variable (i.e. feature) selection in ML. According to (Guyon, & Elisseeff, 2003), those approaches can be classified into three main categories as follows:

- **Wrapper Methods:** Utilising a learning algorithm as a black box to score subsets of variables according to their predictive power.
- **Filter Methods:** Selecting subsets of variables as a pre-processing step, independently on the chosen predictor.
- **Embedded Methods:** Performing variable selection in the process of training, usually with respect to given learning algorithms.

Many algorithms resorted to variable ranking as the main mechanism for feature selection due to its simplicity, scalability, and good empirical success as well. Examples are (Caruana, & Sa, 2003; Forman, 2003; Weston, Elisseeff, Schölkopf, & Tipping, 2003).

Principal Component Analysis

Principal Component Analysis (PCA) is a linear dimension reduction technique (Jolliffe, 1986; Jackson, 1991). The PCA method is based on the covariance matrix of the variables. Basically, PCA performs a linear mapping of data into a lower dimensional space by finding a few orthogonal linear combinations (i.e. PCs) of the original variables with the largest variance. The objective is to obtain a smaller set of orthogonal projections along the original feature set such that the variance of data with the new dimensions is maximised. For many datasets, the first several PCs explain most of the variance with minimal loss of information. PCA has been successfully used in problems of high dimensional datasets, such as in pattern recognition for example (Sirovich, & Kirby, 1987; Turk, & Pentland, 1991).

Linear Discriminant Analysis

A similar approach to PCA is the Linear Discriminant Analysis (LDA) (Fisher, 1938). LDA searches for vectors in the underlying data space that best discriminate among classes, rather than those that best describe the data as in PCA. Specifically, LDA creates a linear combination of independent features which yields the largest mean differences between the desired classes. More formally, two measures are defined as below (Martínez, & Kak, 2001):

1) Within-class scatter matrix, as given by:

$$S_w = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (x_i^j - \mu_j)(x_i^j - \mu_j)^T$$

Where x_i^j is the ith sample of class j, μ_j is the mean of class j, c is the number of classes, and N_j is the number of samples in class j.

2) Between-class scatter matrix, as given by:

$$S_b = \sum_{j=1}^{c} (\mu_j - \mu)\,(\mu_j - \mu)^T$$

Where μ represents the mean of all classes.

Reinforcement Learning

A quite different strategy of ML is termed as *Reinforcement Learning*. The central idea of reinforcement algorithms is to learn optimal actions through trial and error. Unlike unsupervised and supervised algorithms, those algorithms do not focus on representation or prediction, but instead attempt to take optimal actions given the current state of the system. The learning agent decides the best next action based on its current state, and by learning actions maximises the reward. For instance, a robot can learn to avoid collisions by receiving positive feedback (i.e. reward), or negative feedback (i.e. penalty) after avoiding or running into obstacles. The robot can then use that feedback to understand the optimal state of movement and choose the next action. Figure 2.12 sketches the environment of reinforcement learning.

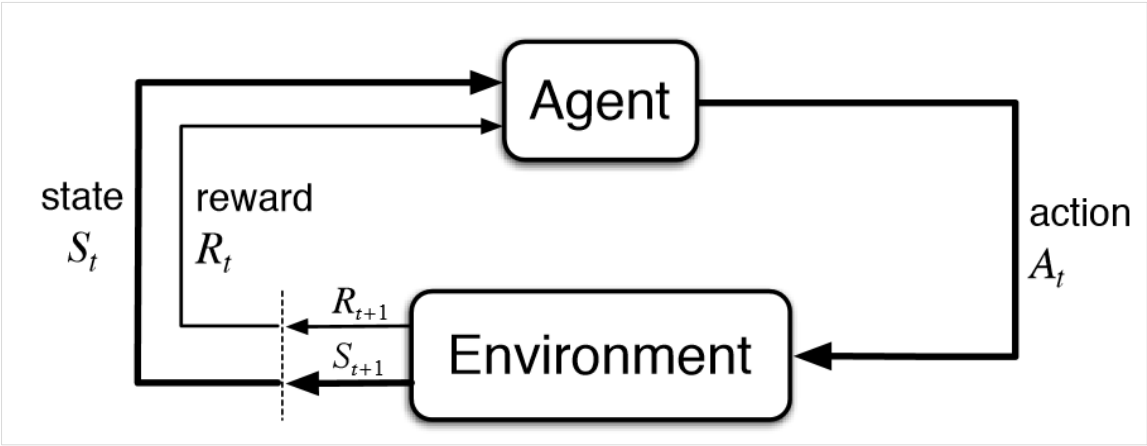


Figure 2.12: Agent in reinforcement learning.

In contrast to supervised learning where training examples are as (input, label), the examples in reinforcement learning come in the form of (input, some output, grade for this output) (Abu-Mostafa, Magdon-Ismail, & Lin, H.T, 2012). (Kaelbling, Littman, & Moore, 1996) defined a formal model of reinforcement learning as follows:

- A discrete set of environment states (S).
- A discrete set of agent actions (A).
- A set of scalar reinforcement signals, typically (0, 1), or real numbers {0, 1}.

2.3 Analytics Tools and Technologies

In this study, a set of tools and technologies were used for the purpose of data pre-processing, and developing predictive models including simulations or ML. This section overviews those tools from a practical standpoint.

Overview of Data Analytics

The opportunities enabled by Big Data have led to a significant interest in the practice of data analytics. Thus, data analytics has evolved into a vibrant and broad domain that incorporates a diversity of techniques, technologies, systems, practices, methodologies, and applications.

Various definitions were developed in order to describe the emerging field of data analytics. Table 2.3 presents a set of common definitions used to describe analytics. In the same context, Figure 2.13 portrays the inter-disciplinarity involved within the practice of data analytics. According to (Mortenson, Doherty, & Robinson, 2015), these disciplines can be fitting into one or more of the following categories:

- **Technological:** Refers to the various tools including hardware, software, and networks, which altogether support the efficient processing of large-scale datasets.
- **Quantitative Methods:** Refers to the applied quantitative approaches to analysing data, such as statistics, ML, econometrics, and OR.
- **Decision Making:** An inherently interdisciplinary area including tools, theories, and practices used to support and understand the decision making process (e.g. human–computer interaction and visualisation in information systems, or problem structuring methods in OR/MS).

Table 2.3: Common definitions of analytics.

Definition of Analytics	Reference
The extensive use of data, statistical and quantitative analysis, explanatory and predictive models, and fact-based management to drive decisions and actions. The analytics may be input for human decisions, or may drive fully automated decisions. Analytics is a subset of what has come to be called Business Intelligence.	(Davenport, & Harris 2007)
Delivering the right decision support to the right people at the right time.	(Laursen, & Thorlund, 2016)
The scientific process of transforming data into insights for making better decisions.	(Liberatore, & Luo, 2011)

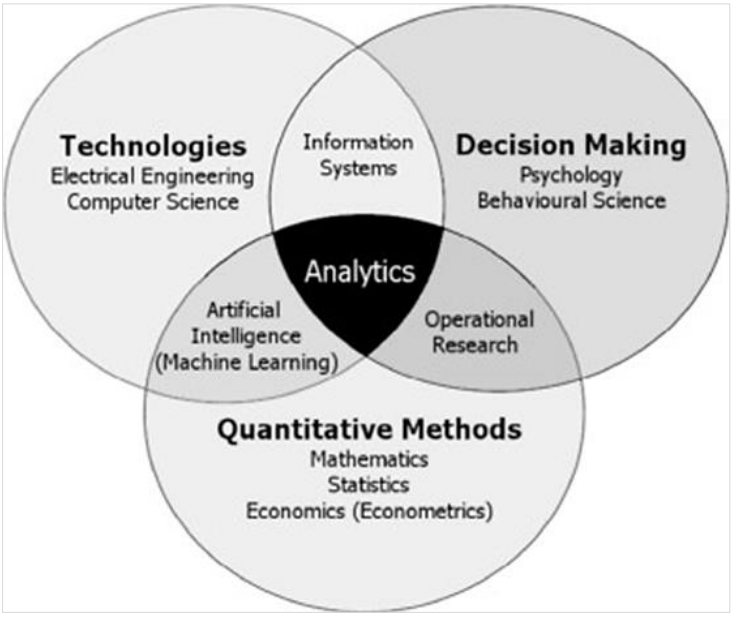


Figure 2.13: Disciplines related to analytics (Mortenson, Doherty, & Robinson, 2015).

Azure Machine Learning Studio

Azure ML Studio provides an ideal environment for data analytics, and ML in particular. With an interactive visual workspace, it becomes easy to build, test, and deploy predictive models. Predictive models can be constructed by dragging and dropping datasets and analysis modules onto the design surface. Perhaps more importantly, ML Studio facilitates to iterate on the design of predictive models in terms of editing parameters or modules, and running experiments multiple times.

Predictive models can be built as *Experiments* in ML Studio. An experiment is made of the components necessary to build, test, and evaluate a predictive model. Those components can be mainly divided into two categories: i) Datasets, and ii) Modules.

A dataset contains data that has been uploaded to Azure ML Studio. Several sample datasets are also provided. A module is an algorithm that can be used while building models. Azure ML studio provides a large set of modules to support the end-to-end Data Science workflow.

Modules include a diversity of functionalities including:

- Normalisation, grouping, and scaling of data.
- Computing statistical distribution of data.
- Conversion to other ML formats.
- Import of data used for experiments and export of results.
- Text analytics, feature selection, dimensionality reduction, and more.

Each experiment can be represented as a complete workflow with all the components required to build, test, and evaluate a predictive model. With a visual representation, ML modules are connected together with lines that show the flow of data and parameters through the workflow. Viewed this way, ML experiments can be easily visualised and explored, and even shared. For the purpose of demonstration,

Figure 2.14 gives an example of an experiment in Azure ML. The experiment starts with loading a couple of dataset named as 'diabetic_data.csv' and 'admissions_mapping.csv'. The experiment avails of the *Join Data* module to merge the two datasets. Afterwards, the combined dataset goes through a sequence of pre-processing procedures including cleaning missing data, removing duplicate rows, splitting data. Eventually, ML model is trained using the *Linear Regression* module. Table 2.4 lists the core modules used in Azure ML Studio (Barga, Fontama, Tok, & Cabrera-Cordon, 2015).

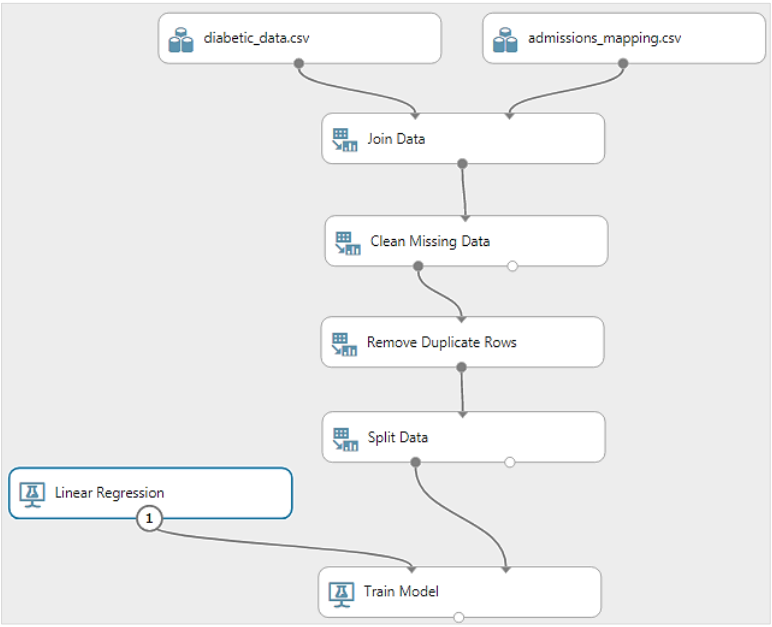


Figure 2.14 : The visual representation of ML experiments in Azure ML studio.

Table 2.4: Core modules in Azure ML Studio.

Module Name	**Functionality**
Convert to ARFF	Converts a .NET serialized dataset to ARFF format.
Convert to CSV	Converts a .NET serialized dataset to CSV format.
Reader	Reads data from several sources including the Web, Azure SQL Database, Azure Blob storage, or Hive tables.
Writer	Writes data to Azure SQL Database, Azure Blob storage, or Hadoop Distributed File System (HDFS).
Moving Average Filter	Creates a moving average of a given dataset.
Join	Joins two datasets based on keys specified by the user. It does inner, left outer joins, full outer joins, and left semi-joins.
Split	Splits a dataset training and test datasets.
Filter-Based Feature Selection	Finds the most important variables for modelling. It uses seven different techniques (e.g. Pearson Correlation, Chi Squared, etc.) to rank the most important variables from raw data.
Elementary Statistics	Calculates elementary statistics such as the mean, standard deviation.
Linear Regression	Create a predictive model with a linear regression algorithm.
Train Model	Trains a classification or regression algorithm with a training dataset.
Sweep Parameters	Finds parameters that result in the best trained model.
Evaluate Model	Evaluates the performance of a classification or regression model.
Cross Validate Model	Performs cross-validation to avoid over-fitting. By default, this module uses 10-fold cross-validation.
Score Model	Scores a trained classification or regression model.

Furthermore, Azure ML provides modules to allow for integrating R or Python scripts into ML experiments through modules called *Execute R Script* and *Execute Python Script*. The execute-script modules enable to specify inputs (i.e. datasets), and an R/Python script. After the module processes the data, it produces a result dataset and an R/Python device output (see Figure 2.15). Currently, Azure ML supports CRAN R 3.1.0, Microsoft R Open3.2.2, and Python 3.5/2.7.7.

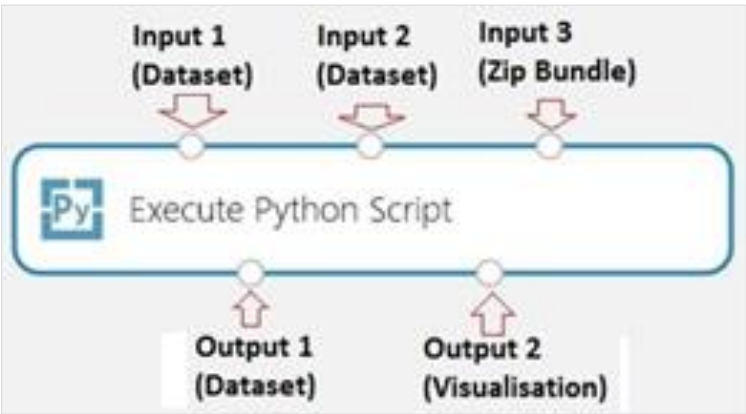


Figure 2.15: The inputs/outputs of Execute-Python-Script module.

Azure Predictive Web Services

The process of deploying ML models in production often involves rewriting the model to run on the target production platform, which can be costly and requires considerable time and effort. In this regard, Azure ML simplifies the deployment of ML models through an integrated process in the cloud. Once deployed, a predictive model runs as a web service that can be seamlessly called from different platforms or programming environments. Only a couple of steps are required to deploy a model in production as follows:

1. Deploying model to staging in Azure ML Studio.
2. In Azure Management portal, moving the model from the staging environment into production.

With the Azure Web service, an external application can communicate with a deployed ML model in real time. A web service call returns prediction results to the external application. The ML web service is based on the REST architecture web programming projects. Azure ML has two types of services:

- **Request-Response Service:** A low latency service that provides an interface to the stateless models created and deployed from the ML Studio.
- **Batch Execution Service:** An asynchronous service that scores a batch for data records.

R-Language

The R language was intensively used within the study for a diversity of tasks including data wrangling, visualisation, and building simulation models. This section briefly reviews the history and main features of the R language, and the core packages used by the study.

History of R (Peng, 2015)

R is an open-source programming language system for statistical computation and graphics. Among other things, it provides programming functionalities, high-level graphics, interfaces to other languages and debugging facilities. The R language was developed as a dialect of S, which was designed in the 1980s and has been widely used in the statistical community (Ihaka, 1998). S is a language that was developed by John Chambers and other fellows at the old Bell

Telephone Laboratories. John M. Chambers was awarded the 1998 ACM Software Systems Award for S.

One key limitation of the S language was that it was only available in a commercial package, S-PLUS. The R language emerged quite a bit after the development of S. In 1991, R was created by Ross Ihaka and Robert Gentleman in the Department of Statistics at the University of Auckland. In 1993, the first announcement of R was made to the public. Ross's and Robert's experience developing R is documented in a 1996 paper in the Journal of Computational and Graphical Statistics (Ihaka, & Gentleman, 1996).

In 1995, Martin Mächler made an important contribution by convincing Ross and Robert to use the GNU General Public License in order to make R free software. This was a critical decision as it allowed for the source code of the entire R system to be accessible to anyone.

Basic Features of R

The language syntax of R depends largely on the semantics of functional programming languages (FPL). In this class of programming languages, subroutines have the ability to modify or construct other subroutines and evaluate the result as an integral part of the language itself. In particular, this allows for the 'computing on the language' concept, which makes it possible to write functions that take expressions as input. Such way of programming is often useful for statistical modelling and graphics. This is similar to Lisp and Scheme, but in contrast to FORTRAN and the ALGOL family.

An important feature that R shares with many popular open-source projects is frequent releases. Throughout the year, smaller-scale releases are made as needed. The frequent releases and regular release cycle indicate the active development of software, and ensures that bugs are addressed in a timely manner. Indeed, while the core developers control the primary source tree for R, many other developers around the world make contributions in the form of new features, fixing bugs, and more. R has maintained the original S philosophy providing a language that is not only useful for interactive work, but also contains a powerful programming language for developing new tools. This enables the user, who takes existing tools and applies them to data, to slowly but surely become a developer who is actually creating new tools.

Another key advantage is that R's sophisticated graphics capabilities. R enables to create publication quality graphics since the very beginning. With more visualisation packages become available, R continues to reinforce that trend. R's base graphics system allows for very

fine control over every aspect of a plot or graph. Further, newer graphics systems (e.g. lattice and ggplot2) allow for complex and sophisticated visualisations of high-dimensional data.

The primary R system is available from the Comprehensive R Archive Network, known as CRAN. R functionality is divided into a number of packages as follows:

- The *base* R system contains the basic package that is required to run R and contains the most fundamental functions.
- Other packages contained in the base system include *utils, stats, datasets, graphics, grDevices, grid, methods, tools, parallel, compiler, splines, tcltk, stats4*.
- There are also other recommended packages such as: *boot, class, cluster, codetools, foreign, KernSmooth, lattice, mgcv, nlme, rpart, survival, MASS, spatial, nnet, Matrix*.

System Dynamics Modelling with R

In this study, the R's *deSolve* package (Soetaert, Petzoldt, & Setzer, 2010) was used for building SD models. The deSolve package greatly facilitates solving ordinary differential equations (ODE), differential algebraic equations (DAE), and partial differential equations (PDE).

For SD models, the ODE solver in deSolve is used. The key requirement is that the equations of an SD model should be implemented as a function, and this function is called by deSolve. Figure 2.16 illustrates how deSolve creates SD models.

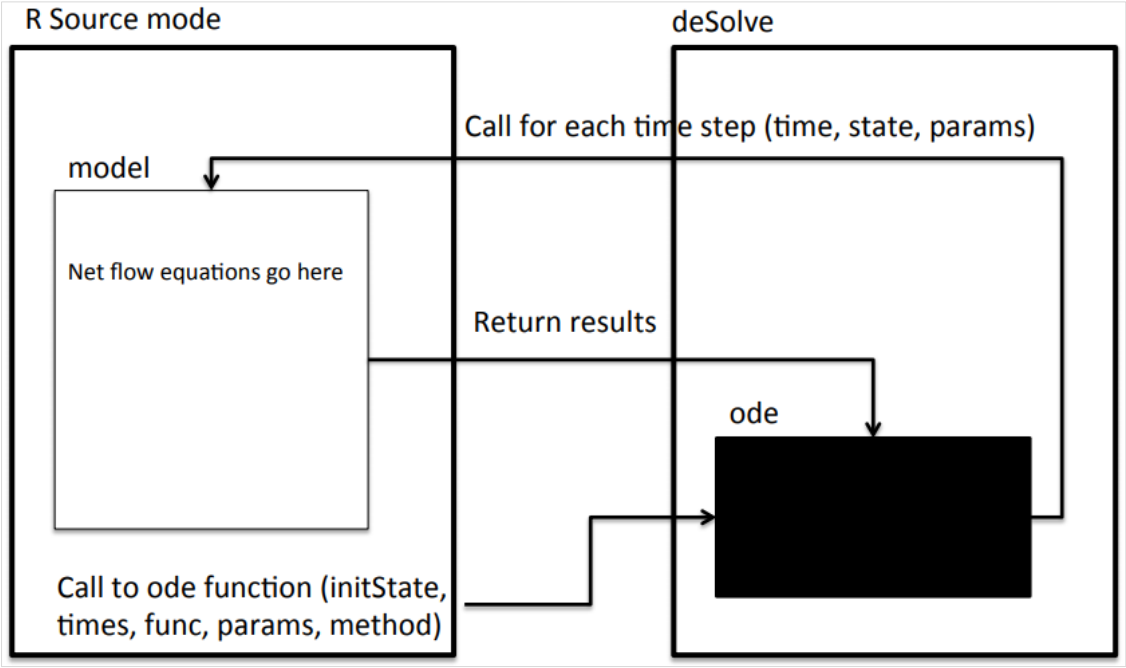


Figure 2.16: How *deSolve* works with SD models (Duggan, 2015).

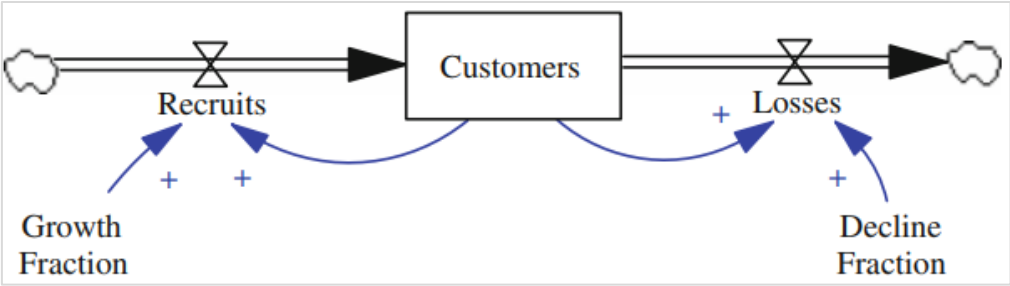


Figure 2.17: SD model of customer growth dynamics (Duggan, 2015).

Figure 2.17 presents an example representing customer growth dynamics from (Duggan, 2016). Through this example, the SD implementation with the deSolve package is described as follows. To use deSolve, the package first needs to be installed, and then it can be referenced in the model source file by calling the 'library()' function.

The first step is to define constants used for the simulation time (i.e. start, finish, and step). The simulation time vector *simtime* is then created using the *seq()* function. The simtime vector partitions the simulated time into discrete intervals of which the simulation model steps one at a time.

Next, two vectors need to be defined, whereas they are required as inputs to the SD model function. The first vector contains the model stocks, along with their initial values. For this simple example, there is only a single stock (i.e. Customers), and its initial value is set to 10000.

The second vector *auxs*, which contains the exogenous parameters of the model (i.e. growth fraction, and decline fraction).

The deSolve expects a function that implements the model equations. That user-defined function will be called from the deSolve library, takes three parameters:

- The current simulation time (time).
- A vector of all current stock values (stocks).
- A vector of model parameters (aux).

These vectors can be transformed to lists using *as.list()*, and embedded in the *with()* function, as this allows the variable names to be conveniently accessed. With these input values, all that remains is to specify the stock and flow equations as follows:

- The flow *fRecruits* is a product of the stock *sCustomers* and the growth fraction *aGrowthFraction*.
- The flow *fLosses* is a product of the stock *sCustomers* and the decline fraction *aDeclineFraction*.
- The net flow for the stock is calculated as the difference in inflow and outflow, and stored in the variable *dC_dt*.

A list structure is then returned to the deSolve package. The first parameter is a vector of all the net flows, and this must match the order in which the stocks are initialised in the vector stocks. Following this, any other model variable can be added to the return list to ensure that appears as part of the final result set. Finally, the model is solved by calling the *ode()* function, which is part of the deSolve library. This function takes five arguments:

- The vector of stocks (y = stocks).
- The simulation time vector (times = simtime).
- The function name that contains the model equations (func = model).
- The auxiliary parameters (parms = auxs).
- The integration method (method = “euler”). Other methods are available, including Runge-Kutta 4th order integration (method = “rk4”).

This data frame can be used as a basis to plot data and also to analyse results. For example, the *summary()* function can be applied to the stock and flows in the data frame, yielding useful summary statistics. Figure 2.18 gives the full R code of the model as explained above.

```
library(deSolve)
START<-2015; FINISH<-2030; STEP<-0.25
simtime <- seq(START, FINISH, by=STEP)
stocks <- c(sCustomers=10000)
auxs <- c(aGrowthFraction=0.08, aDeclineFraction=0.03)

model <- function(time, stocks, auxs){
with(as.list(c(stocks, auxs)),{
fRecruits<-sCustomers*aGrowthFraction
fLosses<-sCustomers*aDeclineFraction
dC_dt <- fRecruits - fLosses

return (list(c(dC_dt),
Recruits=fRecruits, Losses=fLosses,
GF=aGrowthFraction,DF=aDeclineFraction))
})
}

o<-data.frame(ode(y=stocks, times=simtime, func = model,
parms=auxs, method="euler"))
```

Figure 2.18: Example of SD model implementation with *deSolve* (Duggan, 2016).

Discrete-Event Simulation with DESMO-J (Page, & Kreutzer, 2005)

DESMO-J is an object-oriented framework for developing discrete-event simulation models. The acronym DESMO-J stands for 'Discrete-Event Simulation and Modelling in Java'. The full name of highlights two key properties of DESMO-J as:

- DESMO-J is based on the DES approach, where changes in system state are supposed to happen at discrete points in time. The system state is assumed to remain constant between such events. DES models are therefore particularly suitable for systems in which relevant changes of state occur suddenly and irregularly (e.g. queueing networks).
- DESMO-J is implemented in Java. Using this framework to build simulation models ultimately results in writing a Java program.

To support DES modelling, DESMO-J provides components and suitable linguistic abstractions to capture the following concepts:

- Entities that model real-world systems in terms of objects and their properties, behaviour, and relationships.
- Stochastic distributions to model random behaviour of systems.
- Data collectors to record statistical data for analysis.
- A clock entity to track model time.
- An event list to store pending events or process activations.
- A scheduler entity to control model execution, and simulate interactions between conceptually concurrently active entities.

Figure 2.19 sketches the above-mentioned components provided by DESMO-J. The components are also divided into two categories pertaining to model or experiment.

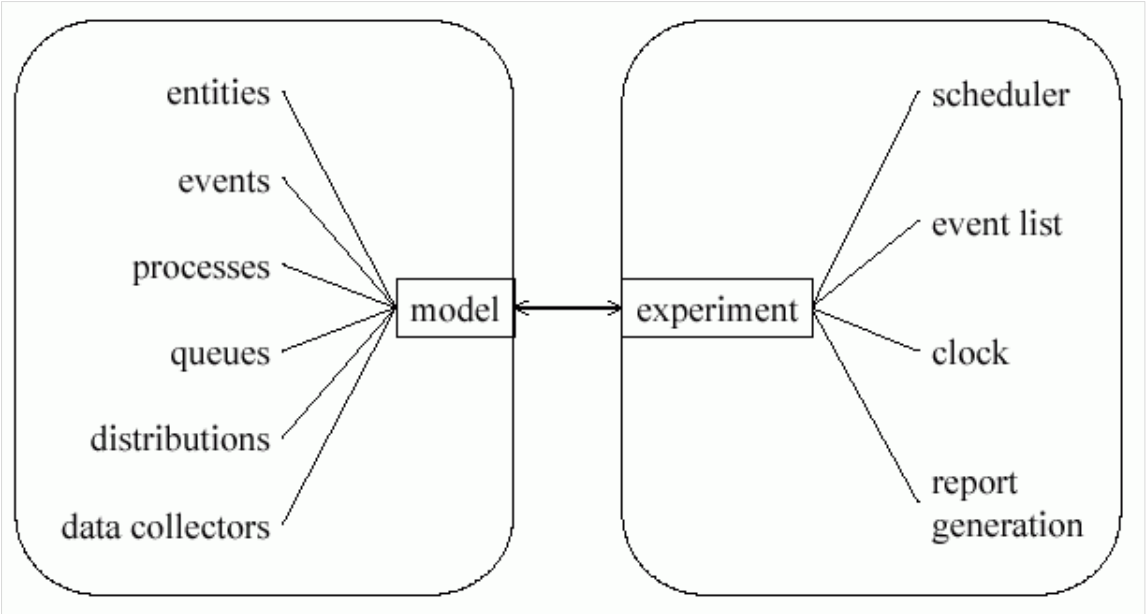


Figure 2.19: Components of DESMO-J (DESMO-J Tutorial, 2017).

DESMO-J separates modelling aspects from simulation experiments. Model properties include declarations and descriptions of entities, events, processes, queues, distributions, and data collectors, while black-box components like scheduler, event list, and model clock are encapsulated by the *Experiment* class. Figure 2.20 illustrates the DESMO-J class hierarchy. The term 'hot spot' refers to those classes whose behaviour needs to be detailed by a model designer (i.e. white box components), while black box components only need to be instantiated and their parameters set. In the figure, white box classes are shaded more lightly than black box classes.

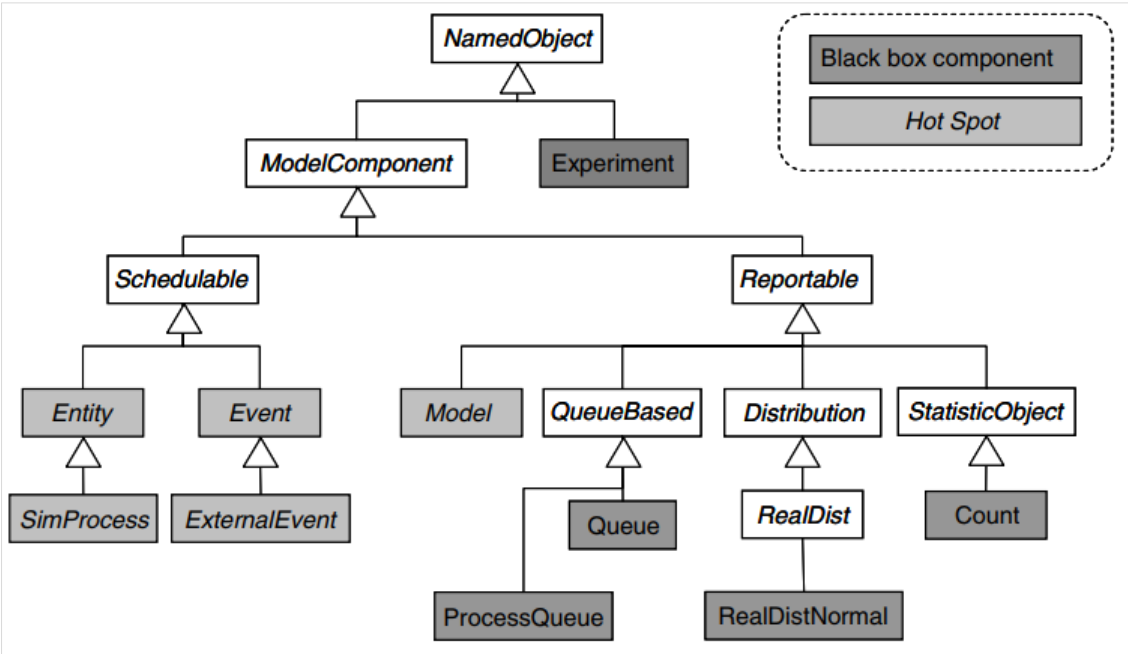


Figure 2.20: The class hierarchy in DESMO-J.

DESMO-J supports different modelling styles including:

- Event-oriented.
- Process-oriented.
- Combined event/process-orientation.
- Activity-orientation.
- Transaction-based modelling.

DESMO-J model construction can be split into a number of tasks as follows:

- Selection of black-box components, statistical distributions, and queues.
- Implementation of lifecycles for dynamic white-box components by customising suitable hot-spot classes (e.g. SimProcess, Entity, and Event).
- Implementation of top level functionality such as instantiation and parametrisation of model components, scheduling of active entities' processes.

The DESMO-J framework is structured into several Java packages (see Figure 2.21) as follows:

- *desmoj.core* provides within its sub-packages all the necessary classes to start modelling with DESMO-J:
    - *desmoj.core.simulator* contains the core classes needed to build a model and set up an experiment. Every simulation model will have to derive a subclass of *Model* which sets up the model and connects it to an experiment. Other classes in this package comprise model components which can either be used directly (e.g. *Queue* or *ProcessQueue*), or sub-classed (e.g. *Entity*, *Event*, and *SimProcess*).
    - *desmoj.core.dist* provides numerous probability distributions with high statistical accuracy. They are based on a linear congruential random number generator as implemented in *java.util.Random*.
    - *desmoj.core.exception* contains classes for the framework's internal exception handling, and can safely be ignored by modellers.
    - *desmoj.core.report* provides DESMO-J's automatic report functionality. The output can be generated in HTML and XML formats.
    - *desmoj.core.statistic* provides a wide range of data collectors, from simple counters to time-weighted mean and deviation of time series data to frequency distributions . All data statistics will be computed and displayed automatically in the end-of-run report.
    - *desmoj.core.advancedModellingFeatures* contains classes to be used in modelling process synchronisation on a more abstract level.

- *desmoj.extensions.applications* contains useful extensions to the core DESMO-J to facilitate model building in several application domains.
- *desmoj.extensions.experimentation* provides a basic graphical user interface to DESMO-J to run experiments and view results.

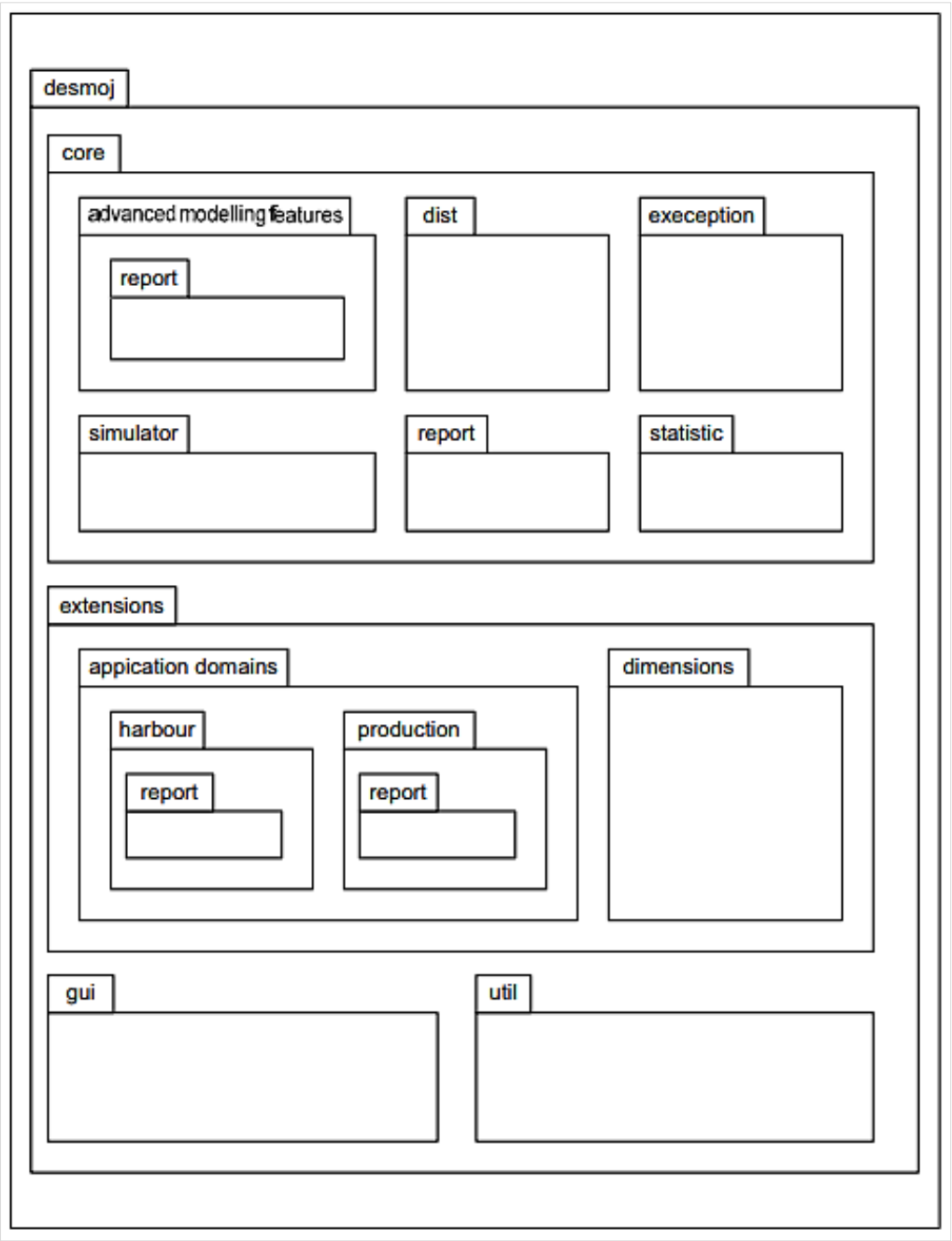


Figure 2.21: The main packages in DESMO-J.

2.4 Application Domain: Healthcare in Ireland

Overview of the Healthcare System in Ireland

A sound simulation study should start with understanding the actual system, or more accurately the problem of interest (Law, 2007). Similarly, the CRISP-DM model (Shearer, 2000) sets the perception of business and data as the initial crucial stage for the process of data mining or knowledge discovery. Therefore, understanding the underlying structure of the Irish healthcare system represented a major aspect of the study prior to building simulation or ML models. This section delivers a basic background to the healthcare system in Ireland, and its underpinning components.

The healthcare system in Ireland has been undergoing a substantial reform based on a phased strategy since 2012. The fundamental goal of the reform is to transition the healthcare system towards the integrated delivery of healthcare services. The integrated care is adopted as a means to improve the services in relation to accessibility, quality and user satisfaction of care services. According to the WHO, integrated care is a concept that brings together inputs, delivery, management and organisation of services related to diagnosis, treatment, care, rehabilitation and health promotion (Gröne, & Garcia-Barbero, 2001).

The transitional arrangements included structuring the Irish healthcare system into 9 geographic regions named as ‘Community Health Organisations’, commonly known by its acronym CHO. The CHOs can be likened to the Accountable Care Organisations (ACO) in the US healthcare system (Gold, 2015). Similarly to the ACOs, the establishment of CHOs aimed to provide coordinated care for patients, such that they get the right care at the right time. To put it in more detailed words, the CHOs are aimed to serve as integrated service areas that can deliver better, more integrated and responsive services to people in the most appropriate setting. Figure 2.22 sketches the phased development of Ireland’s healthcare system towards realising integrated care.

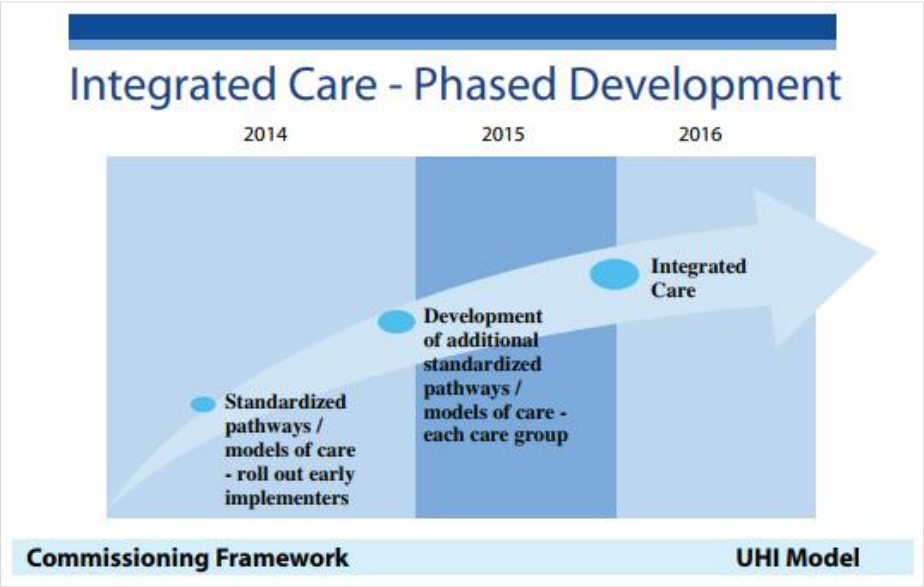


Figure 2.22: The development phases of healthcare system in Ireland (HSE, 2015).

Every CHO is responsible for the delivery of primary and community-based services within national frameworks responsive to the needs of local communities. Specifically, the CHOs include 90 Primary Care Networks (PCNs) across the country, where each PCN is intended to serve an average population of 50K inhabitants. The quality of care provided within a PCN highly depends on how healthcare staff are organised in a way that promotes teamwork to responsively address needs of local people. Figure 2.23 shows the geographic boundaries of the 9 designated CHOs.

Figure 2.23: The geographic boundaries of CHOs (HSE, 2015).

Elderly Healthcare and the Incidence of Hip Fractures in Ireland

In tandem with climate change and global terrorism, the UN identified population ageing as one of the three main global challenges (UN, 2007). Moreover, the acceleration of ageing is likely to increase over the coming decades (Lutz, Sanderson, & Scherbov, 2008). In Europe, the proportion of people aged 65 years and over has already exceeded that younger than 15 years in 2008, and that proportion is expected to double by 2060 (Rechel et al., 2013). More importantly, the proportion of very elderly, aged 80 years and over, is expected to triple between 2008 and 2060 (European Commission, 2009).

Likewise in Ireland, the population has been experiencing a pronounced transition of ageing. The Health Service Executive (HSE) of Ireland reported in 2014 that the increase in the number of people over 65 is approaching 20K per year (HSE, 2014). Population ageing is therefore expected to have profound impacts on a broad range of economic and social areas. Figure 2.24 plots the trend of ageing worldwide as reported by (UN, 2015).

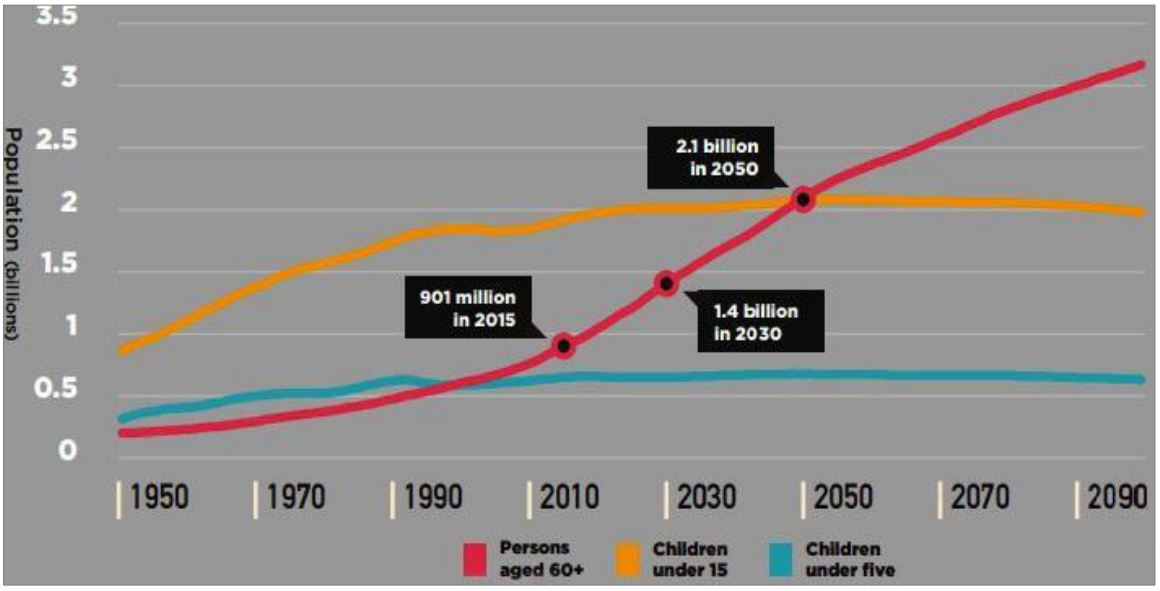


Figure 2.24: Global projections of the elderly aged 60 and above (1950- 2090).

In the context of elderly-related care, the study focused the attention on the care scheme of hip fracture in Ireland. Hip fractures are a major cause of injuries and morbidity among elderly patients. Numerous studies such as (Cooper, Campion, & Melton, 1992), (Melton, 1996), and (Gullberg, Johnell, & Kanis, 1997) recognised hip fractures to be exponentially increasing with age, despite the existence of rate variability from country to another. Furthermore, the burden of hip fractures on the healthcare system may unavoidably increase owing to the continuous improvement of life expectancy of the population (Melton, 1993), and (Kannus et al., 1996).

In Ireland, around 3K people sustain hip fractures annually in Ireland (Ellanti, 2014). Specifically, the rates of fracture for the total population aged 50 years and over were reported as 407 and 140 per 100K for females and males respectively (Dodds, Codd, Looney, & Mulhall, 2009). It was also reported that about 80% of the elderly patients are over 75 years of age (Laffoy, 2008). Therefore, these figures can inevitably increase owing to the growing trend of ageing as shown in Figure 2.25, which plots projections of elderly population in Ireland from 2016 to 2026.

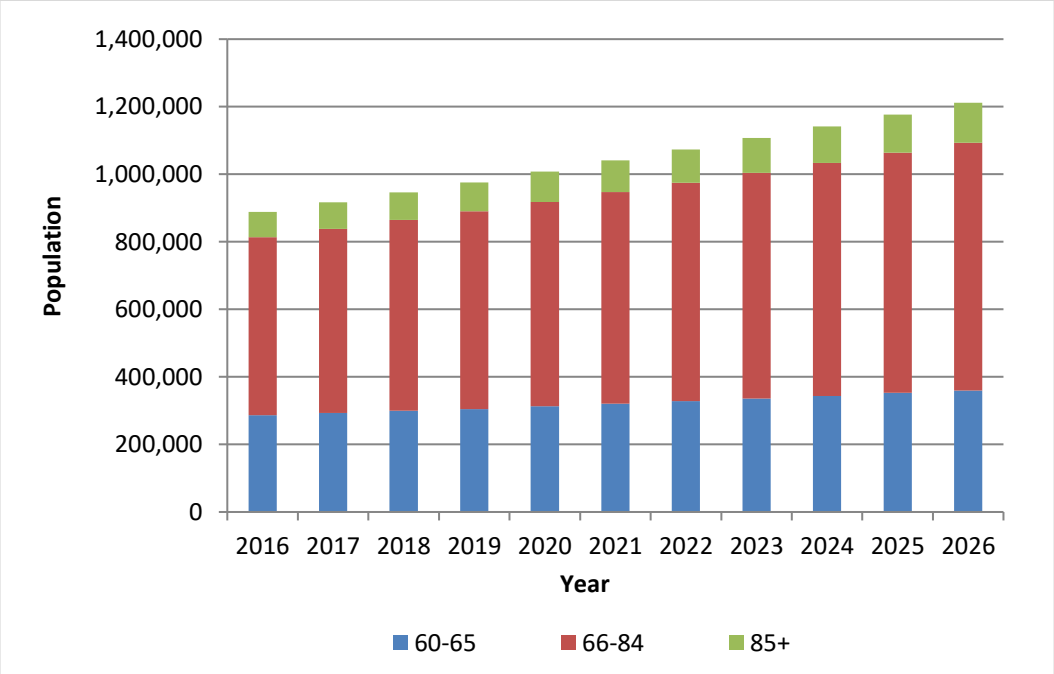


Figure 2.25: The projections of elderly population in Ireland (2016- 2026) (CSO, 2017).

From an economic perspective, hip fractures can represent a major burden on the Irish healthcare system. According to the HSE, hip fractures were identified as one of the most serious injuries resulting in lengthy hospital admissions and high costs (HSE, 2008). The median LOS was recorded as 13 days, and less than one-third go directly home after their hospital treatment (Ellanti, 2014). As a result, plentiful studies attempted to investigate the costs associated with hip fracture incidents including (McGowan et al., 2013), (Azhar, 2008), (Gannon, O'Shea, & Hudson, 2007), (Cotter et al., 2006), (Carey, & Laffoy, 2005), and (Haentjens, Lamraski, & Boonen, 2005). In this regard, the cost of treating a typical hip fracture was roughly estimated around €12K - €14K, and even this figure should probably have increased. In light of that information, the incidence of hip fractures represents a major concern to healthcare in Ireland, and there will be a critical need to develop evidence-based strategies in order to meet the foreseen challenges.

2.5 Data Description

The availability of high-quality information and data was of high significance to realise the study goals. On one hand, quality data is a key determinant of the validity and credibility of a simulation model (Law, 2007). On the other hand, ML is purely a data-driven method, and the learning process would not simply happen without data.

Initially, we inspected a number of data repositories involved in capturing information through the treatment scheme of hip fracture. Three data sources were identified including: I) Irish Hip Fracture Database (IHFD), II) PCT-based data (Primary Care Teams), III) National Incident Management System (NIMS). However, it was only feasible to acquire a dataset from the IHFD repository. Thus, the main source of data used by the study came from the IHFD (NOCA, 2017).

The IHFD repository was developed to serve as the national clinical audit that captures care standards and outcomes for hip-fracture patients in Ireland. The IHFD was initially developed through collaborative efforts among several organisations including the Irish Gerontological Society, the Irish Institute for Trauma and Orthopaedic Surgery, the Healthcare Pricing Office (HPO), and the HSE. Currently, the National Office of Clinical Audit (NOCA) has taken over the governance of the IHFD, and the responsibility for producing the annual reports.

The IHFD archives abundant information about the patient's journey from admission to discharge. Specifically, a typical patient record includes 38 data fields about a variety of information including patient characteristics (e.g. gender, age), and care-related factors (e.g. date of admission, LOS). A thorough explanation of the data fields is available via the official data dictionary (HIPE, 2015). Also, the study *Appendix (I)* includes descriptions of the data fields used within the study.

According to the National IHFD 2015 Report (NOCA, 2015), 16 hospitals have been contributing to the IHFD repository. As shown in Figure 2.26, the included hospitals cover the main geographic regions of the country. The figure also shows that a large proportion of hospitals is scattered around Dublin, the capital city, where the larger part of Ireland's population live.

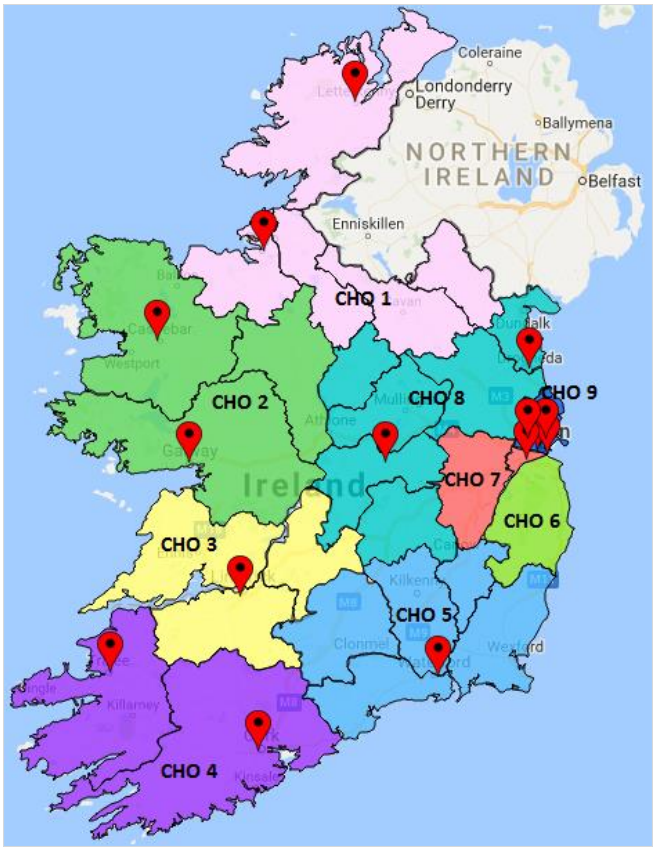


Figure 2.26: Hospitals contributing to the IHFD data.

We acquired a subset of the IHFD that covered three years from January 2013 to December 2015. The dataset included about 8K records with a specific focus on elderly patients aged 60 and over. It is worth mentioning a couple of points pertaining to the dataset. First, the dataset was not evenly divided up among the CHOs. It turned out that the CHOs along the eastern side of the country included the largest part of the patient records. Other areas such as CHO1, CHO2, and CHO4 had relatively less number of records in the dataset. Table 2.5 lists the number of patient records per CHO, while Figure 2.27 visualises the geographic distribution in this regard. Second, a single patient may have been related to more than one record, in case of recurrent fractures. However, it was not feasible to estimate the proportion of recurrent cases, as patients did not have unique identifiers, and records were fully anonymised for the purpose of privacy.

Table 2.5: Counts of records per CHO.

CHO	Records Count / Percentage
CHO1	664 (≈8.5%)
CHO2	570 (7.3)
CHO3	748 (9.6)
CHO4	704 (9)
CHO5	1056 (13.5)
CHO6	869 (11.1)
CHO7	864 (11)
CHO8	1424 (18.2)
CHO9	928 (11.9)

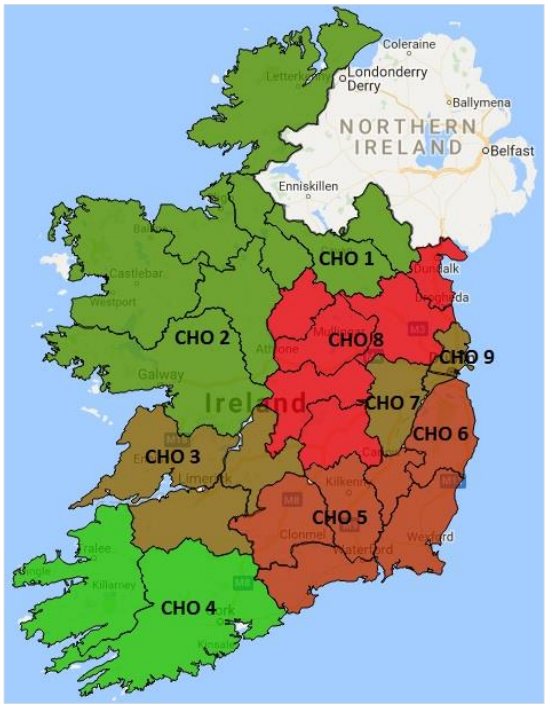


Figure 2.27: Geographic distribution of the dataset. The distribution ranges as a colour gradient between red (high) and green (low).

Figure 2.28 plots a histogram of the age distribution in the IHFD dataset, while Figure 2.29 disaggregates the percentages of age groups with respect to male and female patients. The patients aged 70-89 obviously represented a majority of about 72% of the whole dataset. Similarly, the female patients were considerably more numerous with a percentage of 71% (see Figure 2.30).

Figure 2.31 shows the percentages of fracture types in the dataset. The intracapsular fractures (including displaced and undisplaced types) were more common with 41%, while the subtrochanteric fractures represented only about 7.6%. To get the reader more acquainted with the different types of hip fractures, Figure 2.32 portrays the hip anatomy along with the percentages of those types in the dataset. Finally, Figure 2.33 plots the histogram of the inpatient LOS in the dataset.

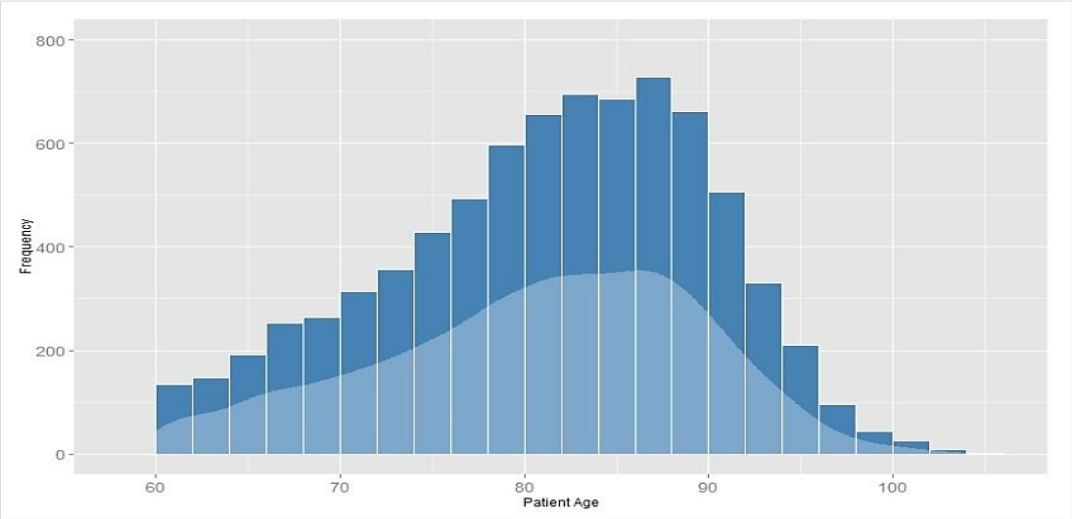


Figure 2.28: The distribution of patient age in the dataset.

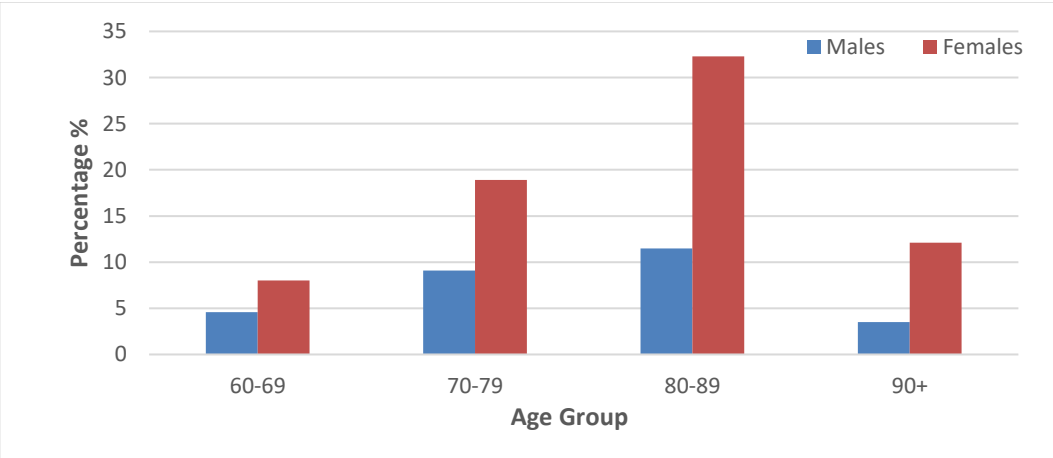


Figure 2.29: Percentages of age groups in males and females.

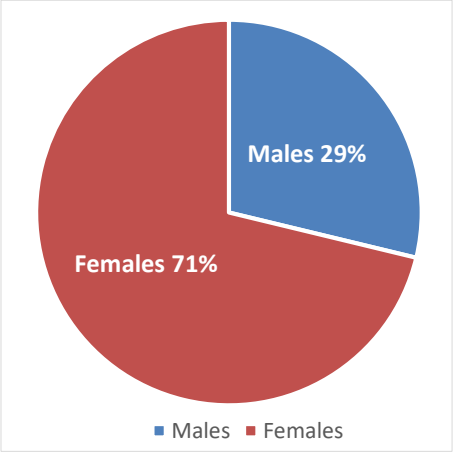


Figure 2.30: The gender percentages of patients in the dataset.

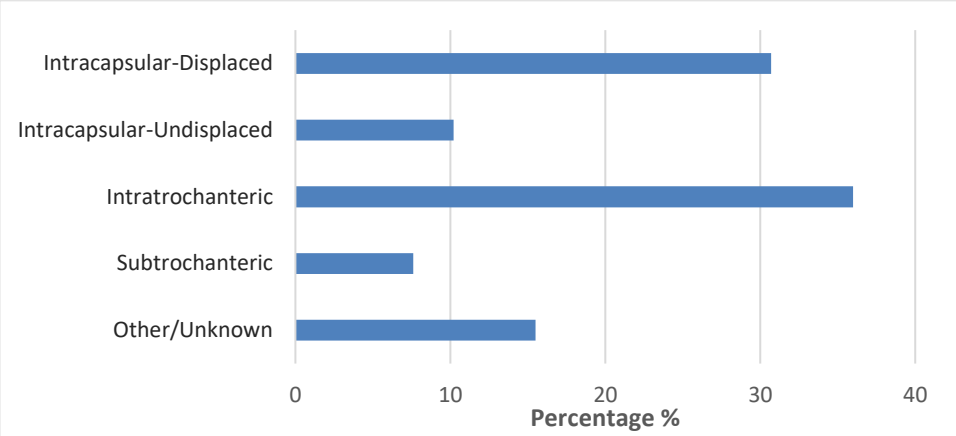


Figure 2.31: Percentages of fracture types in the dataset.

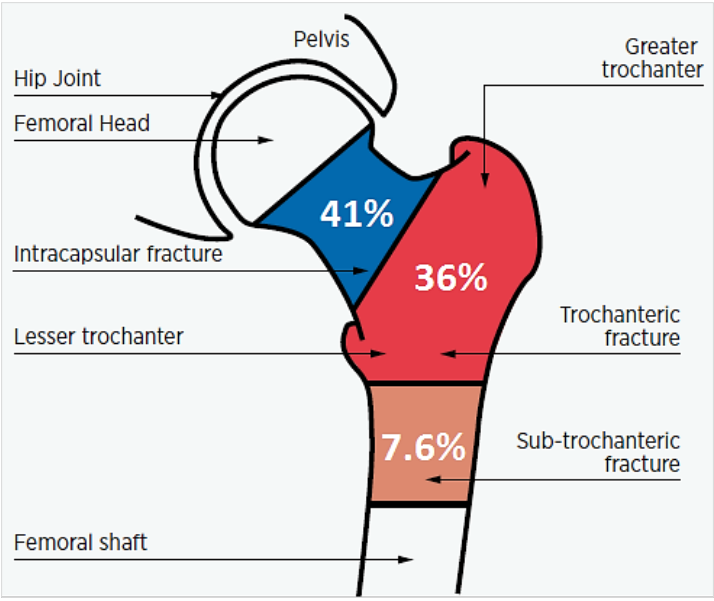


Figure 2.32: The percentages of hip fractures in dataset, adapted from (Parker, & Johansen, 2006).

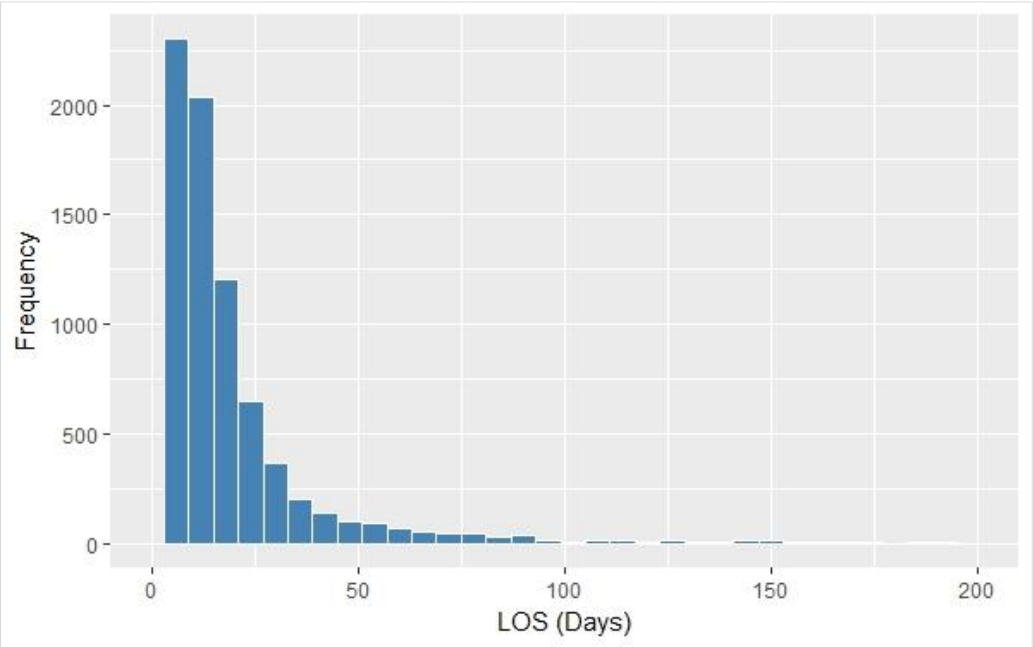


Figure 2.33: The distribution of LOS within the dataset.

Furthermore, the study used population information prepared by the Central Statistics Office of Ireland (CSO, 2017). The data contained comprehensive information about the population in specific geographic areas in terms of age and sex. However, we focused only on the elderly population aged 60 years and over, in line with the study scope. The study also acquired additional census data from the HSE Health Intelligence. The data included comprehensive demographic information mapped to CHOs. The data included comprehensive demographic information mapped to the CHOs. That demographic information was of considerable benefit in order to characterise patient profiles on a population basis. More information on the bed capacities of nursing homes in Ireland was obtained using reports from the Health Information and Quality Authority (HIQA, 2016).

2.6 Related Work

Owing to the multi-faceted nature of the presented work, it is believed that the study can be viewed from different perspectives and contexts. Therefore, related studies were reviewed with relevance a set of topics as follows:

- Hybrid simulations.
- AI-assisted simulations.
- Simulation-based healthcare planning.
- Applications of ML in healthcare.

Hybrid Simulations

It is considered that the most relevant context could be the development of hybrid simulations. As suggested by (Powell, & Mustafee, 2014), a hybrid modelling and simulation study refers to the application of methods and techniques from disciplines like Operations Research, Systems Engineering, or Computer Science to one or more stages of a simulation study. In this sense, this study attempted to integrate simulation models with a method from the Computer Science discipline (i.e. ML).

An interesting classification scheme of hybrid simulations was proposed by (Mustafee et al., 2017) as follows. The classification views hybrid simulations as consolidation of paradigms, methodologies, techniques, and tools. Specifically, the four types of hybrid models were referred as Type (A, B, C and D) as follows (see Figure 2.34):

- **Type A:** Multi-Methodology Hybrid Simulation.
- **Type B:** Multi-Technique Hybrid Simulation.
- **Type C:** Multi-Methodology, Multi-Technique Hybrid Simulation.
- **Type D:** Hybrid Systems Model.

The study's endeavour towards integrating simulations and ML is conceived to fall under the category of Type D. This type of hybrid models was characterised with the combined application of simulation with wider OR techniques that can be applied to one or more stages of a simulation study. This was previously referred to as hybrid M&S study by (Powell, & Mustafee, 2017).

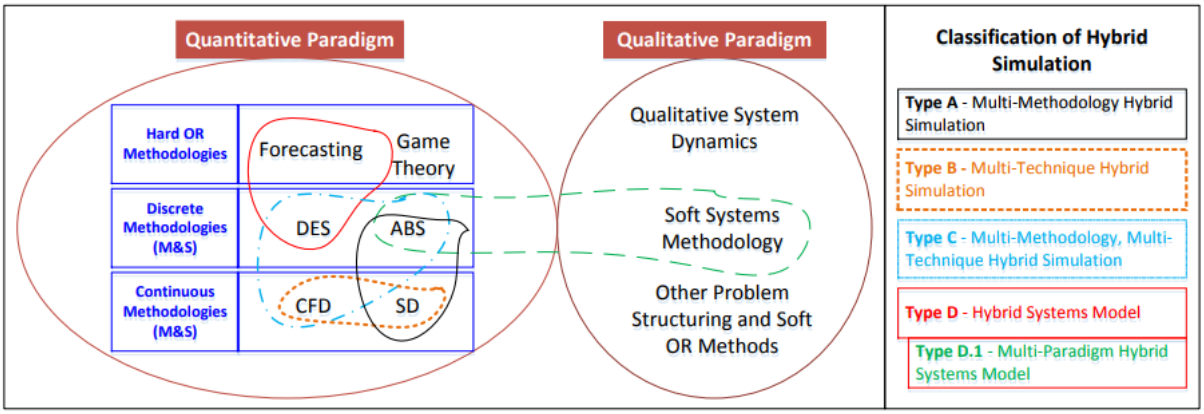


Figure 2.34: Classification of hybrid simulations (Mustafee et al., 2017).

Viewed this way, it was aimed to review examples of hybrid studies that incorporated simulation methods with ML techniques in particular. To focus the search, the study reviewed studies over the past 10 years (i.e. 2007-2016). The majority of studies reviewed came from: i) Winter Simulation Conference, and ii) SIGSIM PADS Conference. It is acknowledged that other relevant studies could have been published in other conferences or journals, but it is believed that the selected venues provided excellent, if not the best, representative studies in accordance with the target context.

The benefits of combining computer simulation and ML techniques were discussed by (Monostori, Kádár, Viharosy, & Stefan, 2000) in the context of manufacturing. A more practical study was developed by (Rabelo et al., 2014). They successfully applied hybrid modelling, where Simulation Modelling and ML were used altogether in a use case related to the Panama Canal operations. A set of simulation models was developed in order to make predictions about the future expansion of the canal. This information was further used to develop ML models (e.g. neural networks and support vector machines) to help with the analysis of the simulation model output.

Further studies utilised ML for the purpose of generating simulation models, or meta-models to be used for generating simulations. For instance, (Bergmann, & Stelzer, 2011) aimed to realise the automatic generation of simulation models, where neural networks were used in order to approximate decision rules. The approach was experimented within a use case related to a manufacturing plant. Similarly, (Morin et al., 2015) experimented a set of ML models to generate meta-models for a simulation aimed to model the sawmilling process. The study used ML algorithms including K-nearest neighbours, decision trees, and random forests.

Further attention was drawn within the area of agent-based modelling as well. A number of studies by (e.g. Wolpert, Wheeler, & Tumer, 1999; Wolpert, & Tumer, 1999) addressed the problem of designing large decentralized multi-agent systems (MAS's) in an automated manner. They presented an approach where each agent runs with a reinforcement learning algorithm. In contrast to traditional MAS design, it was discussed how alternative approaches can potentially benefit from ML. For example, it may not be needed to laboriously model the entire system, and in turn this can facilitate scaling up to large-scale systems. This approach was referred as the *COllective INtelligence* (COIN).

Similarly, Rand (2006) presented a framework that considered ML as a component of ABM. The framework described how different ML techniques can be incorporated into agent-based models. The framework consisted of two cycles that involve evaluating input, making decisions and then generating output. In this manner, the ML algorithm is using the ABM as an environment and a reward generator, while the ABM is using the ML algorithm to refine the internal models of the agents.

More recent studies continued that line of research. For example, (Laite, Portman, & Sankaranarayanan, 2016) claimed that the integration of neural networks into agent-based models can provide a better understanding of dynamic agent responses when modelling complex systems. Likewise, study (Zhong, Cai, Luo, & Zhao, 2016) utilised ML within a use case for crowd modelling and simulation. The ML models were used to learn and predict the flow of crowds.

Other studies availed of ML and data mining techniques in a bid to generate knowledge and decision rules from simulation results. For instance, (Feldkamp, Bergmann, & Strassburger, 2015) proposed an approach for applying data mining methods to simulation data in combination with visualisation methods in order to reveal relationships underlying the model behaviour. They particularly used data clustering to divide large amounts of the simulation output into groups of similar performance values. The clusters were analysed through a set of visualisations to perform a visual investigation process of the simulation data. The approach was also successfully implemented within a case study of gold mining facility (Feldkamp, Bergmann, Strassburger, & Schulze, 2016).

Another study developed a methodology that combined simulation, data mining, and knowledge-based techniques for optimising the management of aircraft fleet (Painter, Erraguntla, Hogg, & Beachkofski, 2006). The study was mainly concerned with determining the impacts of candidate aircraft engine maintenance decisions, in terms of life-cycle cost and operational availability. The simulation output was analysed using data mining techniques to understand the system behaviour in terms of subsystem interactions, and the factors influencing life-cycle metrics. The insights obtained were then encapsulated as policies and guidelines towards supporting the decision-making process.

Further interesting studies discussed the possibilities of taking advantage of more sophisticated ML techniques. In a thought-provoking study, Tolk (2015) envisioned that the next generation of simulation models will be integrated with ML, and Deep Learning in particular. The study argued that bringing Modelling and Simulation, Big Data, and Deep Learning together can create a synergy allowing to significantly improving services to other sciences. Similarly, related studies such as (Pruyt, 2014) and (Pruyt, 2017) interestingly turned the attention to the potentials of integrating SD models with Big Data, or other disciplines related to Data Science.

AI-Assisted Simulations

Shannon (1975) had an early insight about integrating Simulation Modeling with AI:

> *"The progress being made in Artificial Intelligence technology opens the door for a rethinking of the simulation modeling process for design and decision support."*

(Miller, Firby, Fishwick, & Rothenberg, 1992) discussed what AI can allow for the development of simulation models. Examples are data dependencies, backward chaining, and abduction can be used to simplify the creation of simulations. The promise of AI-assisted simulations was that the model behaviour can be generated or adjusted by AI techniques. Examples of approaches were developed in this regard including: i) Knowledge-based simulations, and ii) Qualitative simulations.

Knowledge-based simulation was defined as the application of knowledge-based methods within AI to the field of computer simulation (Fishwick & Modjeski, 2012). As such, the knowledge about the model can be expressed as rule-based facts that characterise experimental conditions. Early endeavours of knowledge-based simulation started with building Prolog-based simulation languages such as TS-Prolog (Futó, & Gergely, 1986), and T-CP (Cleary, Goh, & Unger, 1985). A more recent example of knowledge-based simulations is (Lattner,

Bogon, Lorion, & Timm, 2010) who developed an approach for knowledge-based adaptation of simulation models. The approach was used as a means to enable an automated generation of model variants, with a specific focus on automated structural changes in simulation models.

On the other hand, qualitative simulations followed a different path, which was introduced as an attempt to replicate the process of human reasoning (i.e. qualitative variables) into computer simulations. (Cellier, 1991) defined qualitative simulations as evaluating the behaviour of a system in qualitative terms. However, (Fishwick, & Modjeski, 2012) described that the aim of qualitative simulations is not to adopt qualitative methods instead of quantitative methods, but rather to use qualitative methods as an augmentation to qualitative methods. The QSIM algorithm (Kuipers, 1986) was an early qualitative simulation study.

Critique on Literature

All mentioned efforts are viewed as important and interesting attempts that opened new frontiers for integrating simulations and ML. However, the main argument here is that most of the endeavours can be largely described as 'ad hoc' pursuits with no clear methodology. There is quite limited guidance of why, when, and how to integrate ML within different stages of Modelling and Simulation. Literature obviously lacked pragmatic studies that practically demonstrate the integration of simulation models and ML, to the best knowledge of the author. It is believed that the simulation community would need much further studies that encourage and popularise that integration, and its potential benefits.

Simulation-Based Healthcare Planning

From the context of healthcare decision-making, the study can also be related to similar studies that endorsed care planning using simulation methods. There has been a growing need for healthcare planning in order to keep abreast of the challenge of population ageing. Healthcare executives are continuously challenged by making appropriate decisions in response to the changing profile of population.

Furthermore, healthcare services are delivered in complex environments involving interactions among many care providers and stakeholders. In this respect, numerous studies characterised healthcare environments as complex systems (Tan, Wen, & Awad, 2005; Rouse, 2008; Kannampallil, Schauer, Cohen, & Patel, 2011). A complex system can be basically defined as a system comprising a large number of parts that have many interactions (Simon, 1996).

Specifically, the complexity of healthcare systems can emerge from a wide variety of facets as follows:

- Plentiful agents are involved, including government and policy makers, health insurance providers, suppliers, hospitals, and patients.
- Agents are non-homogeneous, where conflicting goals and interests can likely exist.
- Agents tend to have an inherently dynamic behaviour, which increases the likelihood of a systemic chaotic behaviour.
- There is no single central director or controller of the system as a whole.

In this regard, Modelling and Simulation presents as an effective decision support approach that allows stakeholders to conduct experimental scenarios against models that represent real-world systems (Pidd, 2004). (Royston, 1999) described computer simulation as an alternative to 'learning by doing' or empirical research methods. Abundant studies (e.g. Lowery et al. 1994; Lowery et al. 1996; Harper et al. 2004; Brailsford et al. 2005; Eldabi et al. 2009) aimed at identifying the particular profile of healthcare problems and the way modelling and simulation studies should approach them. (Katsaliaki, & Mustafee, 2011) provided an exhaustive review of papers published between 1970 and 2007 on healthcare-related simulation research. Their results presented a classification of the reviewed publications with respect to simulation techniques employed.

From a practical standpoint, the literature is replete with studies applied simulation to a multitude of issues pertaining to healthcare. For instance, (Harper, and Shahani, 2002) used Simulation Modelling for the planning and management of bed capacities within an environment of uncertainty. The simulation model was utilised to help understand and quantify the consequences of planning and management policies. Similarly, (Rashwan, Ragab, Abo-Hamad, & Arisha, 2013) developed a simulation model in order to map the dynamic flow of elderly patients in the Irish healthcare system. The model was claimed to be useful for inspecting the outcomes of proposed policies to overcome the delayed discharge of elderly patients. (Ragab, Abo-Hamad, & Arisha, 2012) used simulation to address elderly care pathways within the Irish healthcare sector. The developed model was used to assess financial and performance issues related to the flow of elderly patients.

However, the literature obviously lacked similar studies with a specific focus on hip fracture care in Ireland, to the best knowledge of the author. In addition, the literature generally laid little emphasis on endeavours that incorporated simulation–based methods and ML.

Applications of Machine Learning in Healthcare

The adoption of ML has proved promising potentials to improve many aspects of healthcare practice. The literature is rife with contributions that utilised ML in this regard. This section provides examples of related efforts that applied ML within the context of healthcare, with a particular focus on data clustering and rule mining. Further, studies in relation to hip fracture care were reviewed as well, whereas the study experiments were conducted in this context.

Applications of Data Clustering in Healthcare

A variety of clustering algorithms have been used in healthcare-related studies for the purpose of discovering homogenous groups of patients. The partitional clustering approach was widely used for that task, using the K-Means algorithm in particular. For instance, the study (Isken, & Rajagopalan, 2002) applied the K-Means to help guide the development of patient type definitions. The patient grouping was utilised for building simulation models of patient flow in hospitals. Similarly, (Sewitch, Leffondre, & Dobkin, 2004) used the K-Means to identify subgroups of patients based on their responses to a set of tests that assesses their perception of health status. The self-organising maps (SOM) was also utilised by other studies (e.g. Ceglowski, Churilov, & Wasserthiel, 2007) to realise the segmentation of patients. Other studies adopted the hierarchical clustering approach as well. For example, (Newcomer, Steiner, & Bayliss, 2011) used agglomerative hierarchical clustering to identify clinically relevant subgroups based on groupings of coexisting conditions.

Moreover, data clustering received further attention within medical imaging applications. For instance, (Vijay, & Subhashini, 2013) addressed the automatic segmentation of brain tumours in MRI images using the K-Means algorithm. Another study by (Wu, Lin, & Chang, 2007), which developed a colour-based segmentation method based on the K-Means clustering as well. The method was applied for tracking tumours in the MRI brain image. (Ng et al., 2006) proposed a methodology for medical image segmentation that incorporated K-Means with the watershed segmentation algorithm. The use of K-Means clustering was claimed to improve the traditional watershed algorithm.

Applications of Rule Mining In Healthcare

Association rule mining is a well-explored method for learning relationships among variables in large databases. The study by (Brossette et al., 1998) was one of the early attempts to make use of association rule mining in the healthcare domain. Rule mining was specifically used to discover patterns in data about hospital infection control and public health surveillance. The association rules represented outcomes, and the confidence of rules was utilised to monitor changes in the incidence of those outcomes over time.

More recent studies were mainly concerned with discovering the factors that have a potential impact on care outcomes. One example is (Nahar, Imam, Tickle, & Chen, 2013), who availed of rule mining to investigate the factors contributing to heart disease. Three rule mining algorithms were experimented including Apriori, Predictive Apriori and Tertius. Based on the rules discovered, it was indicated that females have less risk of coronary heart disease compared to males. Likewise, (Stilou, Bamidis, Maglaveras, & Pappas, 2001) attempted to extract association rules from a database containing records of diabetic patients using the Apriori algorithm. Other studies aimed at finding frequent patterns in genomic data using rule mining such as (Becquet et al., 2002) and (Creighton, & Hanash, 2003) for example.

Machine Learning in the Context of Hip Fracture Care

In relation to the scheme of hip fractures, a number of studies have used ML for developing predictive models. For instance, some studies focused on predicting outcomes in terms of mortality including (Marufu et al., 2016), (Nijmeijer et al., 2016), and (Karres, Heesakkers, Ultee, & Vrouenraets, 2015). For example, (Marufu et al., 2016) developed a logistic regression model to predict the 30-day mortalities among hip-fracture patients after surgery. The study used a dataset obtained from the UK's National Hip Fracture Database (NHFD).

Apart from these efforts, the care scheme of hip fractures is believed to have received scant attention despite its importance within elderly care. It is argued that literature would need more studies to present data-driven insights in this context. For example, the use of rule mining to the hip-fracture datasets has not been studied before, to the best knowledge of the author. Data repositories such as the NHFD and IHFD would allow for further applications of ML and data-driven insights to improve the quality of care, or curbing costs.

2.7 Summary

This chapter laid the foundational background necessary for the rest the thesis chapters. Initially, the chapter provided a brief overview of the main concepts of Modeling and Simulation. The main simulation approaches were outlined including SD, DES, and ABM. The key constructs of conceptual modelling were also briefly discussed. The background overviewed different approaches of ML including supervised learning, unsupervised learning, and reinforcement learning.

The chapter referred to the tools and technologies used within the study. A set of analytics tools for simulation, ML, or generally data analysis were provided. In particular, Azure Machine Learning Studio, R-Language, SD modelling with R, and the DESMO-J library for building DES models.

Furthermore, the background described the healthcare system in Ireland. The structure of the healthcare system, was examined along with the ongoing reform strategy. Equally important, the dataset used by the study was described in detail. Exploratory visualisations were provided to summarise the key characteristics of the dataset.

Eventually, the chapter discussed related studies from different perspectives. In particular, the literature was reviewed from the following contexts: i) Hybrid Simulations, ii) AI-Assisted Simulations, iii) Simulation-Based Healthcare Planning, and iv) Applications of Machine Learning in Healthcare. The chapter was designed in a way to properly set the scene to the upcoming chapters.

Chapter 3

A Conceptual Framework for Integrating Simulation Models with Machine Learning

3.1 Introduction

The ever-rising complexity of real-world problems and the abundance of information call for further utilisation of machine intelligence to assist the practice of Modelling and Simulation. In a related context, ML has gained a significant momentum as an instrumental artefact for constructing new or improving existing knowledge. With such advances, the study aimed to embrace a path towards exploring the possibilities for simulation models to be supported or integrated with ML.

The main goal was towards realising simulation models that can adjust their behaviour with minimal human input, if any. The initial premise was that the behaviour of a simulation model can be designed or adjusted by the guidance of ML models, which are being incrementally trained to predict the behaviour of the system of interest. In this manner, simulation models can learn to change their behaviour in accordance with changes in the actual system.

This chapter can be viewed as structured into two main parts as follows. The first part spurs a discussion on the prospective integration of simulation models and ML. Subsequently, a conceptual framework is presented to guide the implementation of that integration. At its core, the integration approach is based on the premise that the system knowledge can be (partially) captured and learned in an automated manner aided by ML. It is conceived that the approach can help realise simulation models that learn to change their behaviour in response to behavioural changes in the actual system under study.

3.2 Integrating Simulation Modelling and Machine Learning: The Purpose, Mechanism, and Benefits

This section discusses key constructs related to the potential integration of Modelling and Simulation with ML. This preliminary discussion is intended to set the scene for the rest of the thesis by presenting the study's perspectives on three aspects: the purpose, mechanism, and benefits of such integration.

The Purpose

First and foremost, there is a need to lay out the rationale behind the integration of simulation models and ML. The principal view is that the increasing complexity of data will continue to fuel such integration to evolve and develop. As hybrid simulations (e.g. SD-DES, SD-ABS) have come into prominence to address the multi-faceted complexity of real-world systems such as examples in healthcare (Djanatliev & German, 2013), and supply chains (Lee, Cho, Kim, & Kim, 2002). Similarly, the Modelling and Simulation arena would need to leverage the capabilities of ML to address the rising complexity and multi-dimensionality of data involved in such systems.

The development of simulation models can rely heavily on data-driven knowledge (i.e. empirical data) to build and validate models. Some scenarios may require extracting that knowledge from sheer amounts of data, or high velocity data streams (i.e. Big Data). This can unavoidably translate into further burdens on the modelling process going beyond human capabilities in many respects. Viewed this way, the complexity of data can essentially represent another dimension of the complexity of systems under study.

In this regard, ML presents as a valid path in case of modelling systems that are inherently encompassed with data of high complexity. With ML, the process of learning about such systems can be possibly automated (fully or partially). Further, it is conceived that ML can work effectively in situations where the system's behaviour to be largely learned by examples, rather than expressed analytically (e.g. equations). For instance, (Zhong, Cai, Luo, & Zhao, 2016) utilised ML within a use case for crowd modeling and simulation. The ML models were used to learn and predict the flow of crowds.

The Mechanism

The mechanism of integration is another necessary point to consider. The initial concern is to determine the stage(s) of model development where ML integration could be applicable. The development of simulation models is an iterative process from conceptual modeling to implementation, as illustrated in Figure 3.1. The study's perspective is that ML can be utilised in different ways at different stages of model development as follows.

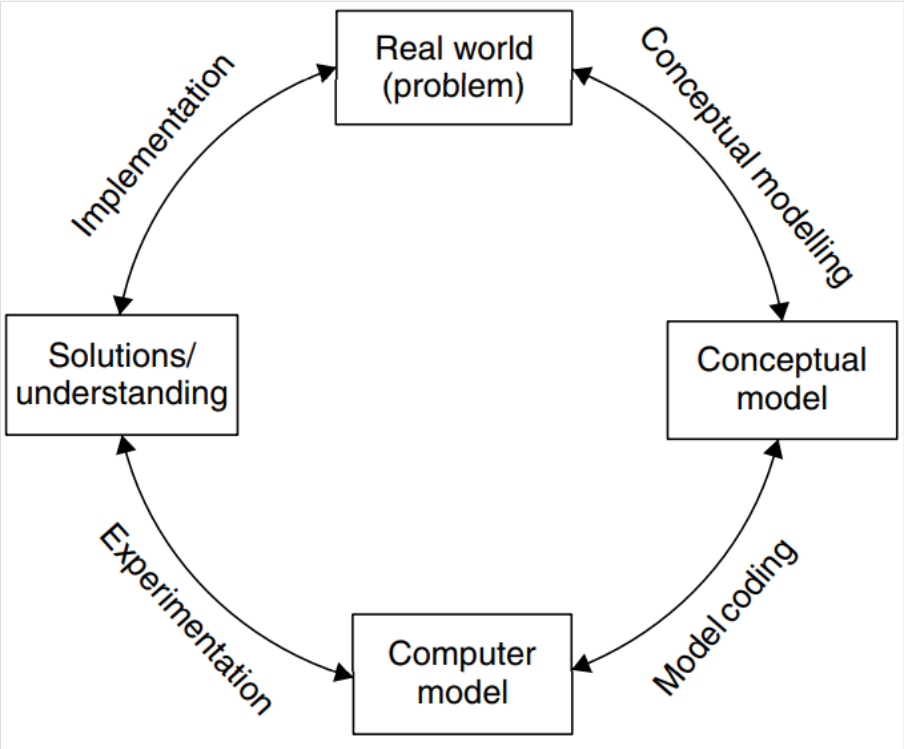


Figure 3.1: Key stages of simulation modelling (Robinson, 2004).

i) Unsupervised ML: System Conceptualisation

In general, unsupervised ML can adequately serve the purpose of knowledge elicitation with minimal prior assumptions. Unsupervised techniques (e.g. clustering, rule mining) can be employed at the phase of conceptual modelling as an assistive artefact for the conceptualisation of the system's structure or behaviour. For example, data-driven insights gained by a clustering model can help identify significant structures or patterns underlying the system of interest. This can in turn reflect on the structure and design of the simulation model. Similarly, unsupervised ML can be used to explore and analyse data output from simulation experiments.

ii) Supervised ML: Guided Simulations

On the other hand, supervised ML techniques can be utilised at the experimentation phase. ML models can be embedded into simulation experiments to realise the learning factor or adaptive behaviour. Concurrently with a simulation experiment, the simulation model can be guided by ML models trained to predict the system's behaviour. Such predictions would suggest the behaviour at the micro-level of the system of interest, as entities or agents in DES or ABS models. Figure 3.2 illustrates the idea of ML-guided simulation experiments in abstract.

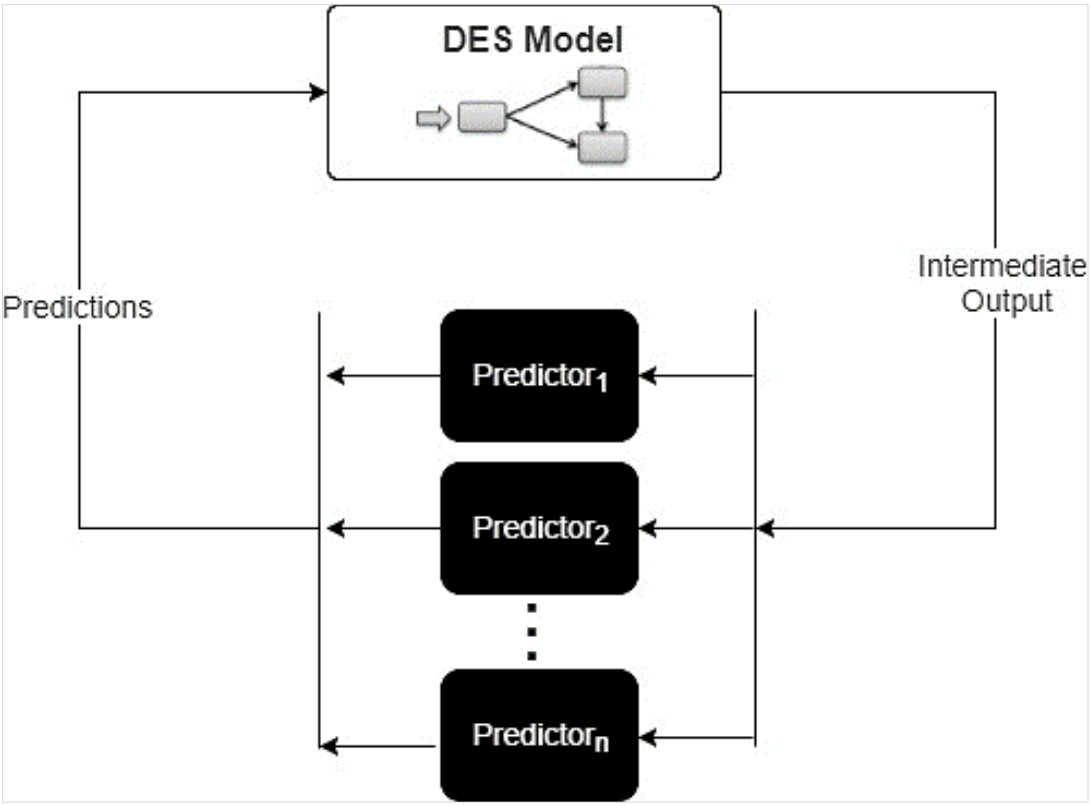


Figure 3.2: ML-guided simulation experiment. The figure shows a DES model that produces intermediate output during a simulation experiment. In conjunction with the experiment, ML models are utilised to make predictions to update the model's behaviour.

The Benefits

The benefits of integration can be viewed from different angles as follows. First, using ML at the conceptualisation process can contribute to lowering the bias of human-based elicitation of knowledge. This can respectively reduce the epistemic uncertainty underlying simulation models (Oberkampf et al., 2002). That kind of uncertainty is largely attributed to the subjective interpretation of systemic knowledge by modellers, simulationists, or subject matter experts.

Further, the application of ML-guided simulations can attain a higher level of model realism. Accurate ML predictions can arguably extend the fidelity of modeling with regard to the micro-level behaviour of simulation entities or agents.

Perhaps more importantly, ML-guided simulations can realise the adaptive behaviour of simulation models. Adaptive simulations can be an attractive approach in case of modeling dynamic systems that inherently exist in rapidly changing environments, where changing conditions can lead to a mismatch between models and actual problems. In this regard, the power of learning from data is that the entire process can be automated with minimal, or without, involvement of human input. This is actually one of the goals that the study attempted to practically demonstrate.

3.3 Approach: Simulation Modelling Aided by Machine Learning

The approach presented in this chapter focuses on integrating simulation models with ML at the experimentation phase to realise an adaptive behaviour. The intention was to develop a conceptual framework that serves as a guide to help this integration to develop in a consolidated manner. The following sections presents the key ideas embraced to develop the approach.

Key Idea I: Learning to Predict the System Behaviour

Starting with a basic view of the components of a typical system. As viewed by (Zeigler, & Sarjoughian, 2012), a system is composed of a set of: i) Structural description, and ii) Behaviour description. A simulation model is constructed based on the combination of those descriptions, as shown in Figure 3.3.

The initial aspect of the study's approach extends the structure of simulation models presented by adding another component for ML models. Generally, it is aimed to utilise ML for predicting the behaviour of the actual system under study. It should be assumed that the system of interest produces sufficient amounts of data that can be used for training ML models.

Based on this hybrid perspective, a simulation experiment is guided by ML models trained to make predictions on the system behaviour. The predictions would aim to reflect the behaviour of system's actors, and how they can react to scenarios in the environment setting in a realistic manner. Figure 3.4 portrays the first key idea in our approach.

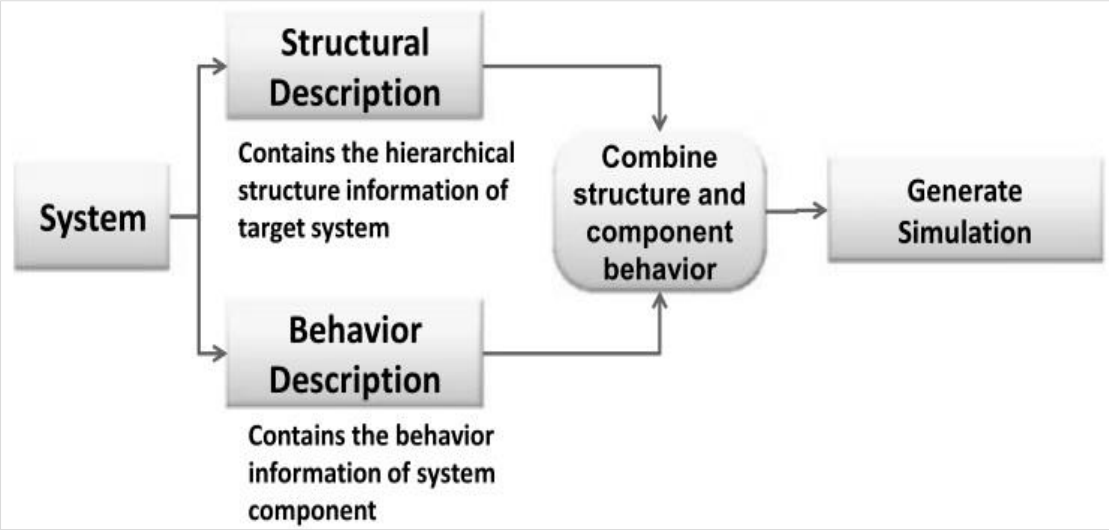


Figure 3.3: Components of a simulation model.

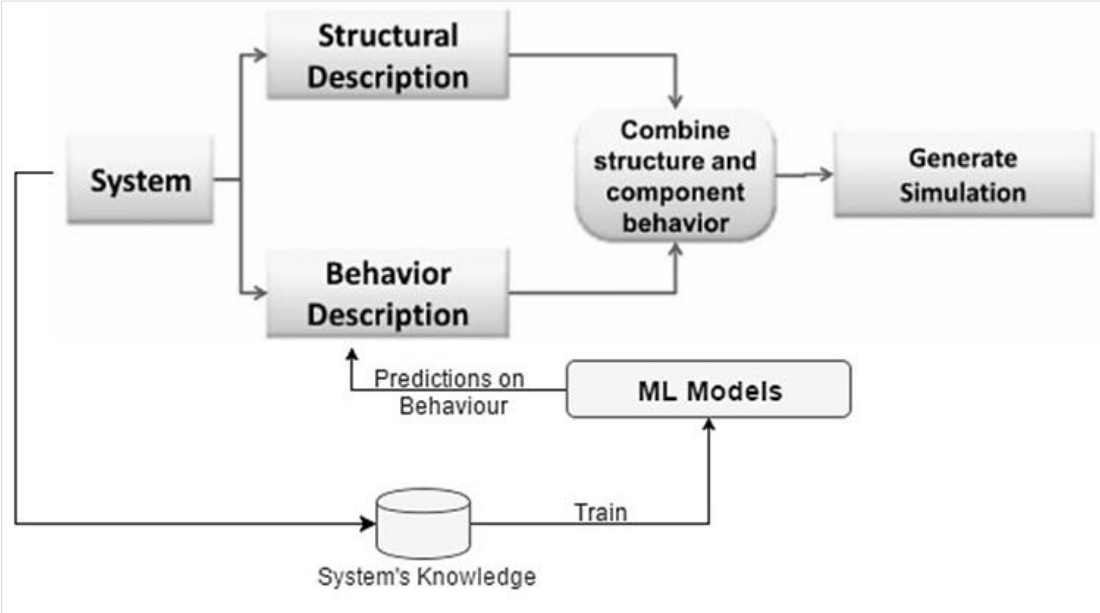


Figure 3.4: Basic view of ML-aided simulations (Key Idea I).

Key Idea II: Identify Predictable Influential Variables

The first key idea briefly discussed how a simulation model could be supported or augmented with ML models. However, it may not seem reasonable to train ML models to predict all variables in a simulation model for many reasons. First, a typical simulation model can include a plethora of variables, so it would be practically infeasible to train hundreds or even tens of ML models. This can significantly overburden the process of developing simulation models.

In addition, further issues have to be considered pertaining to the suitability of ML itself for the problem. As generally recognised in ML literature, a pattern has to exist within data to consider ML as a valid path for building predictive models (Abu-Mostafa, Magdon-Ismail, & Lin, H.T, 2012). Therefore, ML would not be useful in case that the system behaviour largely occurs in a random manner.

To address these issues, the proposed approach follows a simple process for screening variables ahead of building ML models. The screening process is particularly concerned with filtering system variables in terms of: i) Significance with regard to the system behaviour, and ii) Predictability. In other words, we seek for variables that have a considerable influence within the problem context, and can be largely predicted as well. We refer to that category of variables as *'Influential Variables'*, defined as below.

Influential Variable: A variable that has a significant influence on the system under study with respect to the question(s) of interest, whereas the variation of that variable can lead to a change in policy, strategy, or decision-making.

Figure 3.5 outlines the screening process as a flowchart. The initial step is to identify all influential variables. Then, two conditions have to be met. First, there should be enough empirical data for ML. Second, the predictability can then be checked for initially screened variables. Means of data visualisation can be useful in this regard. Different types of plots can reveal promising patterns. If any of the two conditions is not met, then ML is not appropriate in that case.

Subsequently, ML models can be trained and tested using the data available. Eventually, trained models can be integrated with the simulation model. Figure 3.6 updates the approach view by having a definite set of ML models that guide the simulation model.

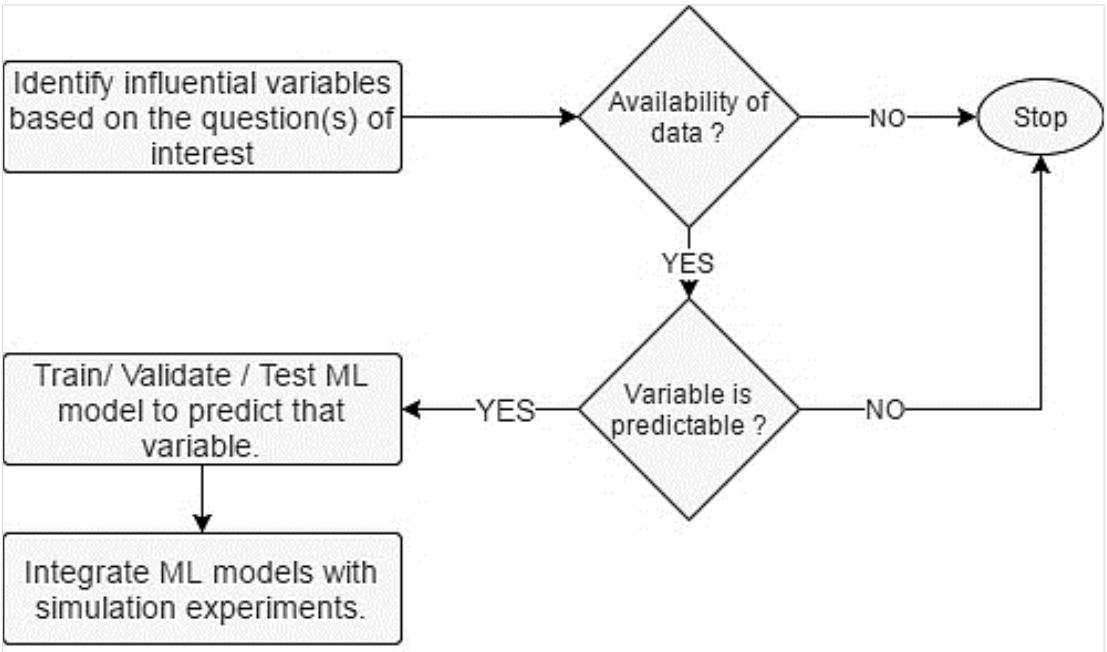


Figure 3.5: The process of developing assistive ML models.

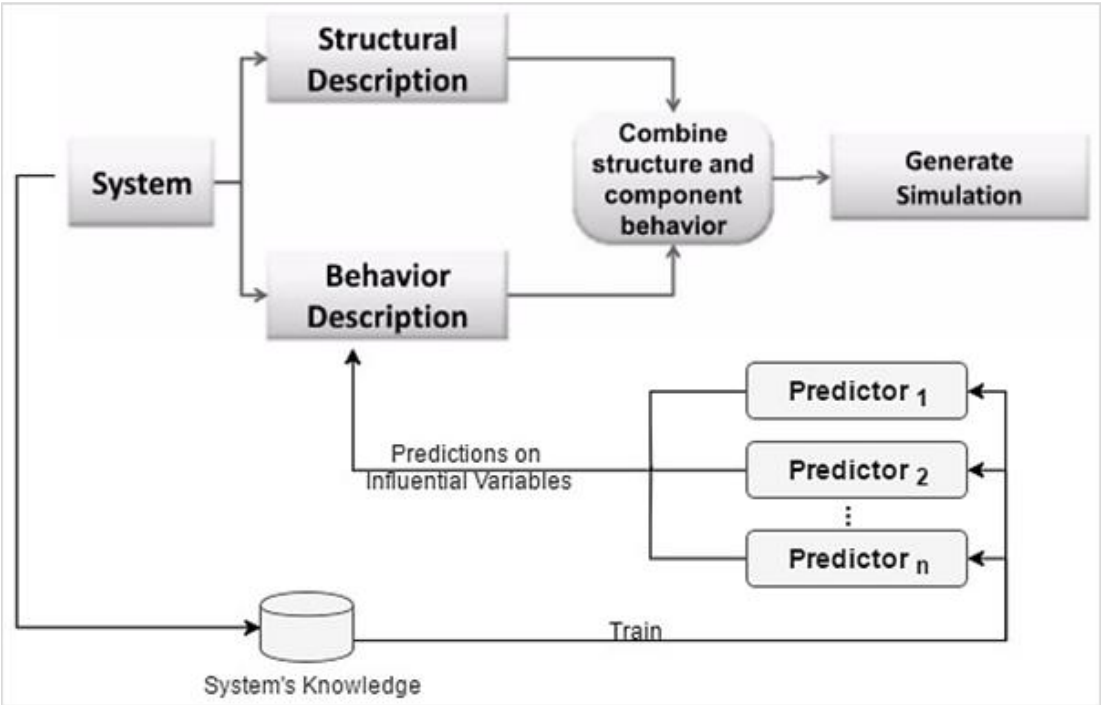


Figure 3.6: A simulation model aided by a set of ML models (Key Idea II).

Key Idea III: Incremental Learning = Dynamic Behaviour

The primary goal of our approach is to realise an adaptive simulation model that can readjust its behaviour with minimal human input. The behavioural adjustment would correspond to changes or new conditions in the actual system. This can only be realised if the ML training is an incremental process, rather than one-off. As such, ML can play an effective role by making the simulation model aware of system knowledge updates.

The idea of incremental learning is based on the premise that new system states are being continuously captured in timely snapshots of data, and added up to an accumulated repository of system knowledge. In this manner, ML models can be iteratively trained in order to learn about possible changes in system behaviour. Figure 3.7 concludes the structure of the approach with the three key ideas explained.

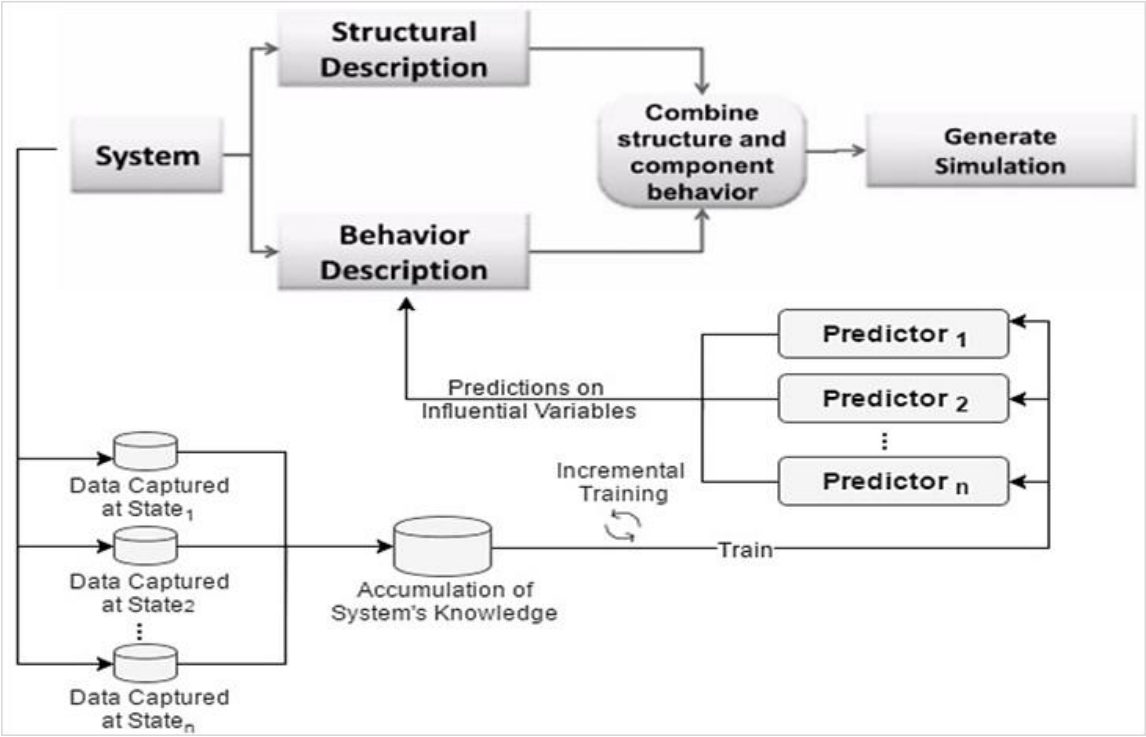


Figure 3.7: Dynamic behaviour using incremental learning (Key Idea III).

The idea of incremental learning can be linked to one of the common concepts within systems modeling, which is feedback loop. As viewed by (Forrester, 1968), the feedback loop (Figure 3.8) is a closed path connecting in sequence a decision that controls action, state of the system, and information about that state returning to the decision-making point.

In this context, the central idea is to consider new data generated by the source system as form of feedback. With such data-driven feedback, ML models can be continuously trained to reflect the system behaviour.

As such, the incremental training of ML models can capture knowledge updates. In turn, changes in the system behaviour can be inferred through ML predictions. Figure 3.9 illustrates the rationale of the approach in a feedback loop-based fashion.

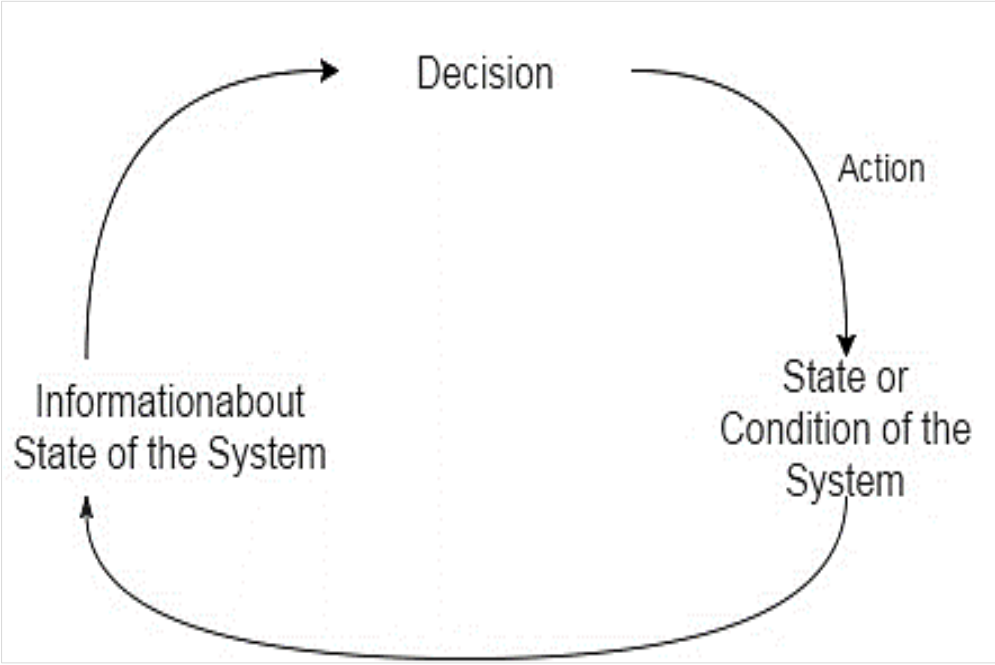


Figure 3.8: The feedback loop.

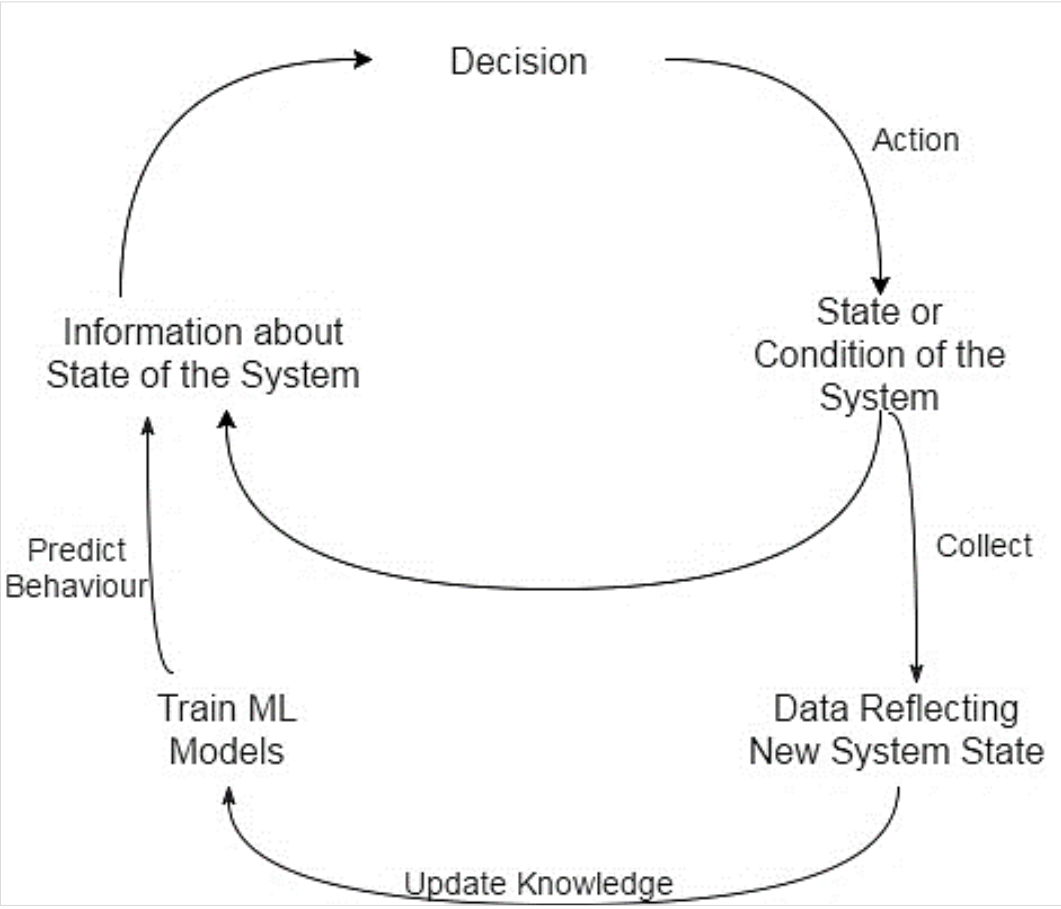


Figure 3.9: Data-driven feedback loop.

3.4 Limitations

A set of limitations are outlined as follows. It is acknowledged that there would be other technical issues pertaining to the implementation of ML-guided simulation experiments. For example, what are the simulation software available that allow for a rapid injection of new states or transitions into a simulation model? More efforts are needed in this regard to prove wider applicability of the proposed approach.

Further, it is conceived that a typical Big Data scenario can better present the benefits of the approach. For example, the case of rapid streams of data would require leveraging the capabilities of ML.

3.5 Summary

The chapter developed a conceptual framework to guide the implementation of integrating simulation models with ML. At its core, the presented approach is based on the premise that system knowledge can be (partially) captured and learned in an automated manner aided by ML. The following chapters will make use of that approach while at different stages of simulation development.

The approach is conceived to help realise adaptive simulation models that can learn to change their behaviour in response to behavioural changes in the system of interest. This can forge a distinctive path for modeling dynamic systems that essentially exist in rapidly changing environments.

Chapter 4

Unsupervised Machine Learning: Knowledge Discovery from Patient Records

4.1 Introduction

The healthcare arena is being substantially transformed by advances in the ability to capture, store, process, and learn from data. Healthcare environments emanate sheer amounts of data from which machine intelligence can provide insights for curbing rising costs and care improvement. In this chapter, it is aimed to discover knowledge from patient records in the IHFD dataset. The intention here was to utilise unsupervised ML techniques in order to learn about patients and the hip-fracture care journey in general. Subsequently, the knowledge gained can be used within building simulation models of the healthcare process. That data-driven knowledge can reflect on the structure or behaviour of the simulation models to be developed in the next chapters.

The study was particularly interested in discovering knowledge in a two-fold aspect. First, it was aimed to investigate whether the dataset inherently contained a tendency of clustering formation. The discovery of patient clusters was believed to extend opportunities for answering questions pertaining to improving the patient journey from admission to discharge. For instance, cluster analysis may reveal possible correlations between patient profiles, care-related factors, and care outcomes. With such correlations, planning procedures could be advised towards improving the quality of care delivered.

Second, it was attempted to discover underlying frequent patterns within the care records patients. Association rules can be derived based on frequent patterns, which can help us to learn factors contributing to care outcomes. For example, a particular combination of patient characteristics and care-related factors may lead to a prolonged inpatient LOS. To focus the study objectives, a set of motivational questions were developed with respect to the directions described above (see Table 4.1).

Table 4.1: Exploratory questions.

Aspect	Motivational Questions
Discovering Patient Clusters	• Does clustering of the data space indicate coherent patient clusters based on specific similarity measures? • If so, how do clusters vary with respect to patient attributes (e.g. age, gender, or fragility history)? • And, how do patients within clusters vary regarding care outcomes (e.g. LOS)? • Further, do clusters reveal possible correlations between care outcomes and other care-related factors such as time elapsed between admission and surgery for example?
Rule Mining	• Are there frequent patterns related to patient attributes, care-related factors, and/or outcomes? • If so, what are the potential factors that can contribute to improving the patient's journey (e.g. avoiding prolonged LOS)?

4.2 Discovering Patient Clusters

"Understanding our world requires conceptualising the similarities and differences between the entities that compose it", Tyron and Bailey (1970).

The initial stage of knowledge discovery aimed to explore latent structures within the dataset. The attempt was to group elderly patients based on the similarity of characteristics, care-related factors, and/or outcomes. The following sub-sections explain the data pre-processing procedures, and how clusters were computed.

Objectives

Data clustering was employed in order to realise the segmentation of patients from a data-driven viewpoint. Clustering was generally defined as the segmentation of a heterogeneous population into a number of more homogeneous subgroups (Aldenderfer & Blashfield, 1984). In this sense, clustering presents as an appropriate method for dealing with voluminous data without making prior assumptions. Clustering techniques are widely used for exploratory data analysis, with applications ranging from statistics, computer science, and biology to social sciences or psychology. Different tasks and purposes can be served by clustering including: i)Exploring the underlying structure of data, ii)Discovering meaningful patterns, and iii)Summarising principal characteristics of data.

According to (El-Darzi et al., 2009), there are four different approaches used for grouping patients into a form of coherent clusters based on: i)Diagnosis-based grouping, ii)Resource consumption-based grouping, iii)Patient pathway grouping, iv)Multi-stage grouping, and v)Clustering-based grouping. This work can be considered to adequately fall into the category of clustering-based grouping.

Clustering Approach

The study embraced the partitional clustering approach using the K-Means algorithm. Due to its conceptual simplicity, the K-Means algorithm has been widely used for clustering tasks. Using a simple iterative technique, the K-Means algorithm can find clusters underlying a dataset, with the number of clusters represented by the variable K. The algorithm iteratively places data points into clusters based on the similarity of features provided. The goal is to minimise the within-cluster sum of squares (WSS) as in the equation below (Jain, 2010). The algorithm converges to a solution when meeting at least one of these conditions: i) The cluster assignments no longer change, or ii) The specified number of iterations is completed.

$$J(C_k) = \sum_{X_i \in C_K} \|X_i - \mu_K\|^2$$

Where μ_K is the mean of cluster C_k, and $J(C_k)$ is the squared error between μ_K and the points in C_k.

Feature Selection

The K-Means algorithm is originally applicable to numeric features only, where a distance metric (e.g. Euclidean distance) can be applied for measuring the similarity between data points. However, it is worth mentioning that there are some variations of the K-Means algorithm that attempted to incorporate categorical features, such as the K-Modes algorithm (Huang, 1998).

In our case, the dataset contained a number of numeric features that can be used effectively. Specifically, the model was trained using the following features: i)LOS, ii)Age, and iii)Time to Surgery (TTS).

Feature Extraction

In a report named as the 'Blue Book' (British Orthopaedic Association, 2007), important guidelines were provided reflecting good practice at key stages of hip fracture treatment. The report defined six quality standards in particular. Among those standards, two quality measures could be captured from the IHFD dataset as follows:

- All patients with hip fractures should be admitted to an acute orthopaedic ward within 4 hours of presentation.
- All patients with hip fractures who are medically fit should have surgery within 48 hours of admission, and during normal working hours.

In light of that, it was expected that those factors can have an influence on care outcomes including LOS, so they can serve as adequate features in the clustering model. However, the raw dataset did not include any fields that explicitly captured such standards. Therefore, they were derived based on the date-time values of patient arrival, admission and surgery. In this way, two new features were added named as 'Time to Admission (TTA)' and 'Time to Surgery (TTS)'. Eventually, only the TTS was included because the TTA contained a significant amount of missing values.

Outliers Removal

Plentiful studies such as (Wu & Yang, 2002), and (Bradley & Fayyad, 1998) have addressed the sensitivity of the K-Means algorithm to noise, or outlier values. Specifically, unusual data points that are significantly distant from the cluster's centroid can have a disproportionate impact on the composition of clusters.

In our case, outliers obviously existed while inspecting the inpatient LOS. The LOS outliers can be observed in Figure 4.1, which plots a histogram of the LOS variable. Those outliers were exceptionally longer than the mean and median LOS values, reported as 19 and 12.5 days respectively (NOCA, 2015). Therefore, we considered only patients of LOS no longer than 60 days in order to prevent the adverse influence of outliers.

In addition, more outliers were observed within the TTS values, where unreasonably large values were observed (e.g. extremely large or negative values). The negative values were altered to the mean value of TTS (i.e. 24 hrs), while other outliers were excluded from the dataset. Overall, the excluded outliers represented about 23% of the dataset.

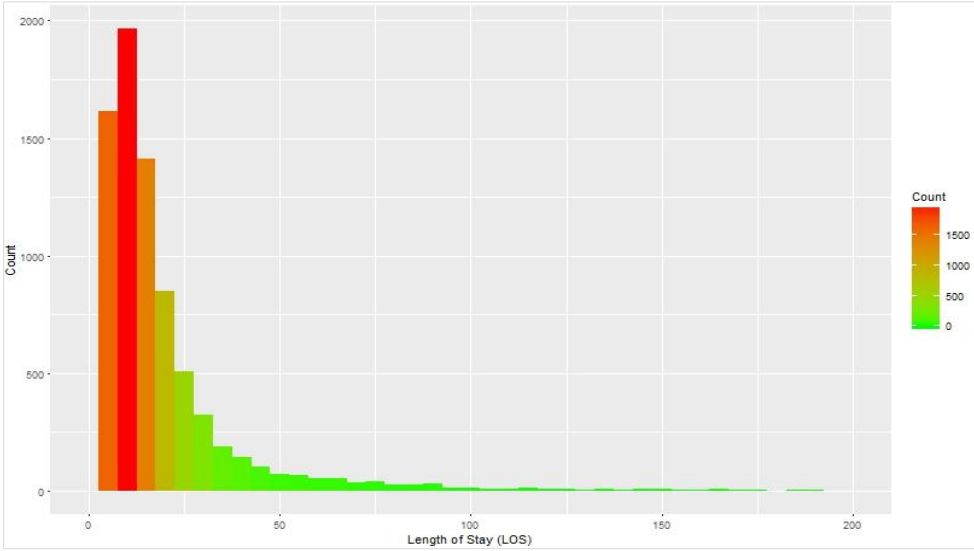


Figure 4.1: Histogram and probability density of the LOS variable. The density of frequency is visually expressed as gradient ranging from green (low) to red (high). The outliers can be obviously observed when LOS becomes longer than 60 days.

Feature Scaling

Feature scaling is a central pre-processing step in ML in case that the range of features values varies widely. A number of studies (e.g. Visalakshi & Thangavel, 2009; Patel & Mehta, 2011) stressed that large variations within the range of feature values can affect the quality of clusters. The min-max normalisation method was used, where every feature was linearly scaled to the [0, 1] interval. The values were transformed using the formula below:

$$z = \frac{x - \min(x)}{[\max(x) - \min(x)]}$$

Clustering Experiments

The K-Means algorithm requires specifying at the outset the number of clusters to be formed. The optimal number of clusters is desired such that adding another cluster does not improve the explained variance to a notable extent. In our case, the clusters were computed using iterations of K-Means with the number of clusters (K) ranging from 2 to 7. Initially, the quality of clusters was examined based on the WSS, as plotted in Figure 4.2. In view of that, it turned out that there could be three or four well-detached clusters of patients that can best separate the dataset (i.e. elbow point). Table 4.2 shows the parameters used in the clustering experiments.

Table 4.2: Parameters of the K-Means algorithm.

Parameter	Value
Number of Clusters (K)	2–7
Centroid Initialisation	Random
Similarity Metric	Euclidian Distance
Number of Iterations	100

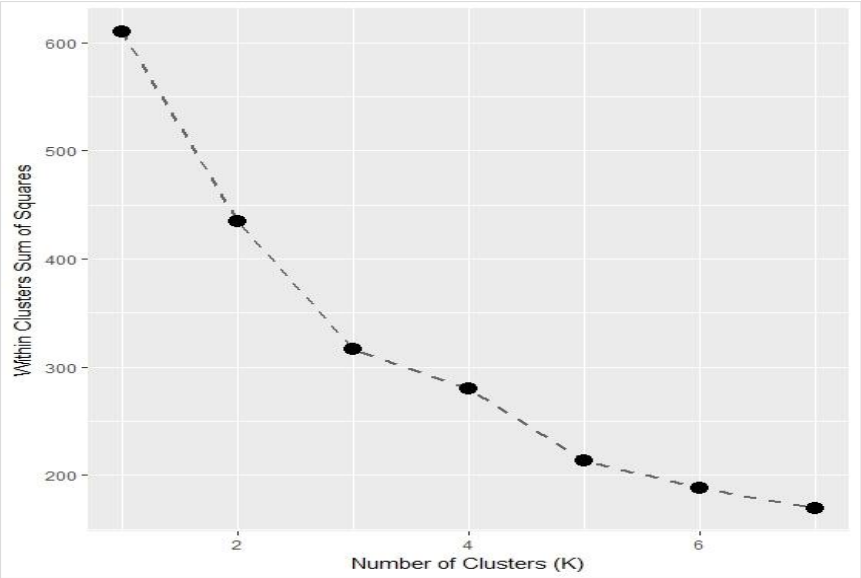
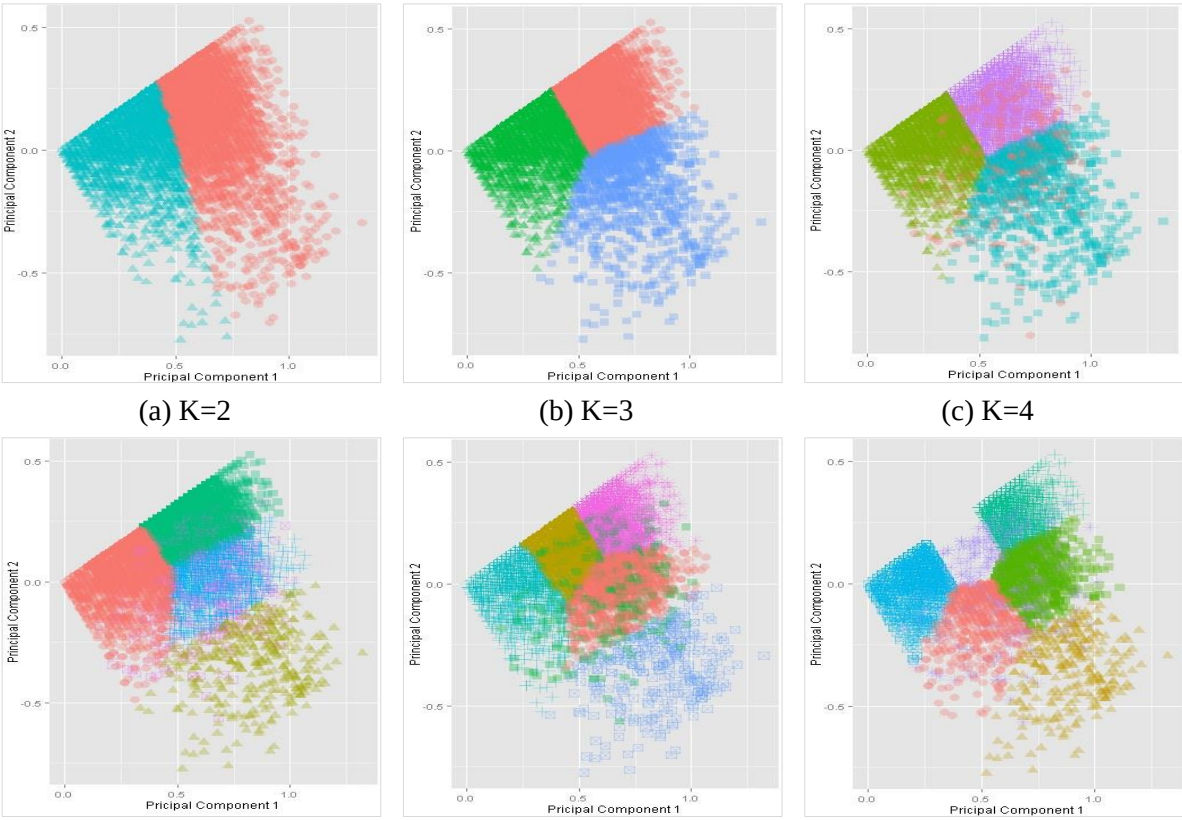


Figure 4.2: Plotting the sum of squared distances within clusters.

To further investigate cluster modularity, the clusters were projected using the Principal Component Analysis (PCA). Figure 4.3 plots the two components that best describe the variance and linear structure of clusters. Each sub-figure represents the output of a clustering experiment using a different number of clusters (K). Initially with K=2, the output indicated a promising tendency of clusters, where the data space is obviously separated into two big clusters. Similarly for K=3, the clusters are still well-separated. However, the separation of clusters started to decline when K=4 onwards. Thus, it turned out eventually that there were three clusters that divided the dataset into coherent cohorts of patients. The cluster visualisations were produced using the R-package ggplot2 (Wickham, 2016).

The experiments were run using the Azure ML Studio. Azure ML provides a convenient cloud-based scalable environment, with the flexibility to integrate with R scripts as well, as explained in Chapter 2. The experiment can also be viewed online through (Azure ML, 2017).



(a) K=2 (b) K=3 (c) K=4

(d) K=5 (e) K=6 (f) K=7

Figure 4.3: Visualisation of clustering experiments with K ranging from 2 to 7. The clusters are projected based on PCA. Overlaps indicate less separation of clusters.

The general characteristics of clusters are briefly presented in Table 4.3. The table compares the count of patients, gender distribution, and other care-related factors (e.g. LOS and TTS) within clusters. Further comparative analysis is conducted in the following sections.

Table 4.3: Standard deviations of the TTS and LOS variables in clusters.

	Patient Count	**Avg. Age**	**Gender Distribution**		**Avg. LOS (Days)**	**Avg. TTS (Days)**
			Males Count	**Females Count**		
Cluster1	2,735	86.78	640	2,095	8.65	1.62
Cluster2	1,111	83.39	363	748	31.27	2.32
Cluster3	2,164	71.67	720	1,444	8.57	1.62

Cluster Analysis

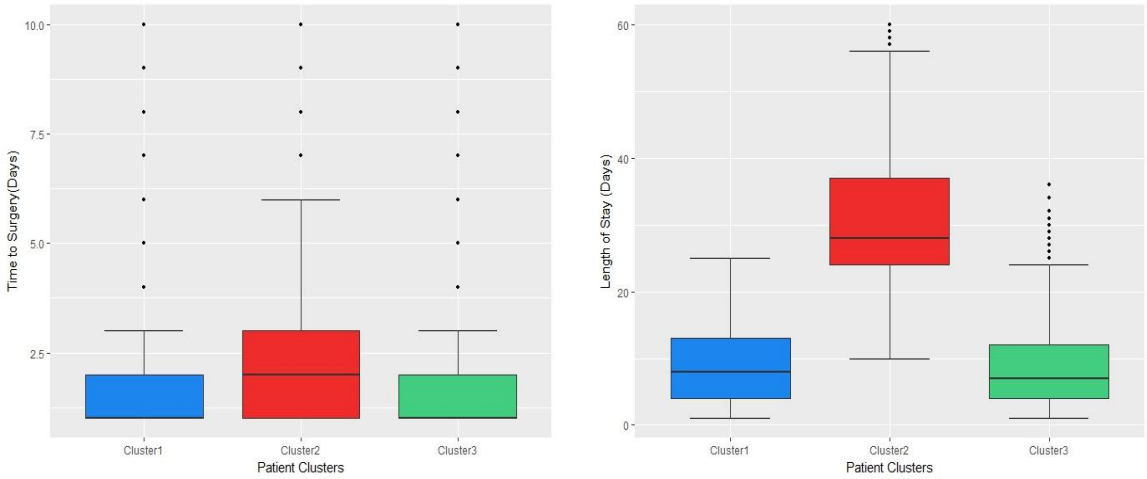
The clusters were explored in a visual manner that can reveal potential correlations or insights. The clusters were examined with respect to the following considerations: i) Patient characteristics (e.g. age, fracture type etc.), ii) Care-related standards (e.g. TTS), and iii) Outcomes (e.g. LOS and discharge destination).

Understanding the factors that have a potential impact on the LOS and discharge destination is of vital importance within healthcare schemes. Recognised by many studies, the LOS was considered as a significant measure of care outcomes (O'keefe, Jurkovich, & Maier, 1999; Englert, Davis, & Koch, 2001). From an operational standpoint, the LOS was considered as a valid proxy to measure the consumption of hospital resources (e.g. Faddy & McClean, 1999; Marshall, & McClean, 2003). In the case under study, it was also reported that the LOS largely accounts for the cost of hip fracture care (Johansen et al., 2013).

Similarly, the early prediction of discharge destinations can be of strategic significance in order to estimate the desired capacity of long-stay care facilities such as nursing homes for example. Moreover, patients discharged to long-stay care facilities are likely to spend prolonged periods of residential care. As a result, the cost of care can significantly increase causing a severe financial burden to the healthcare system.

i) TTS and LOS:

As mentioned before, the TTS is considered as one of the quality standards for hip fracture care. The study therefore explored the clusters regarding the TTS and its possible impact on the inpatient LOS. It turned out that the patients of Cluster2 tended to have a relatively wider dispersion of the TTS compared to Cluster1 and Cluster3. Interestingly, Cluster2 patients also experienced longer LOS compared to Cluster1 and Cluster3, which shared a very similar distribution of the TTS and LOS variables. Figure 4.4 (a) plots the TTS variable in the three clusters, while Figure 4.4 (b) plots the LOS variable. Table 4.4 compares the standard deviations of the TTS and LOS variables in clusters.



(b) TTS (c) LOS

Figure 4.4: The variation of the TTS, and LOS variables within the three patient clusters.

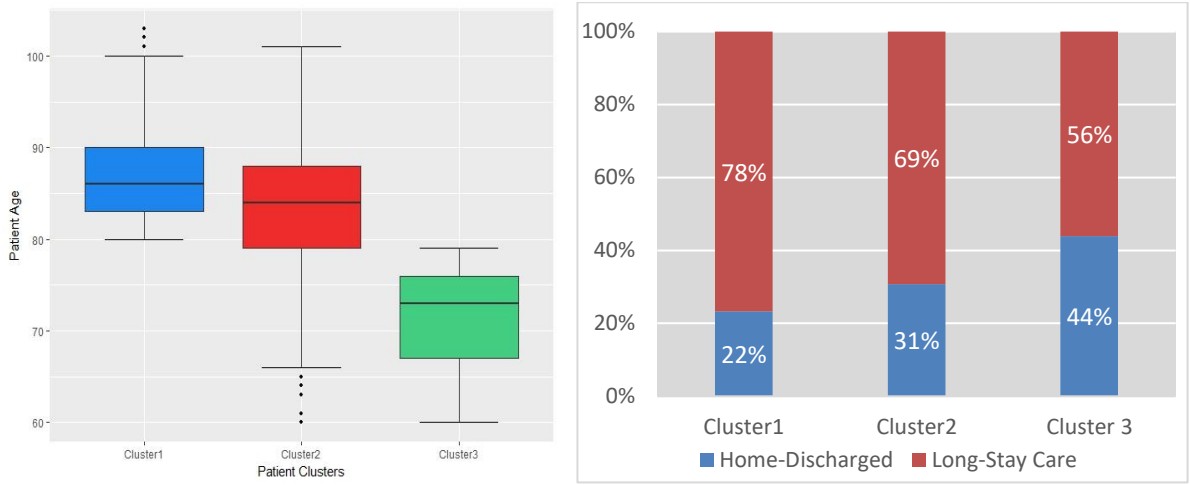
Table 4.4: Standard deviations of the TTS and LOS variables in clusters.

	Standard Deviation (≃)	
	TTS	**LOS**
Clsuter1	1.18	5.78
Clsuter2	1.91	10.63
Clsuter3	1.22	6.22

ii) Age and Discharge Destinations:

Generally, there is a strong emphasis on the patient age in elderly care schemes. In this respect, the possibility of sustaining hip fractures is expected to increase significantly with age as mentioned before. Figure 4.5(a) plots the variations of age with respect to the three clusters. On one hand, it can be observed that only Cluster1 and Cluster2 included all the patients aged 80 or over, while Cluster3 contained only patients aged less than 80 years.

On the other hand, the clusters were inspected with regard to discharge destinations. The discharge destination were broadly classified into: i) Home, or i) Long-stay care facility (e.g. nursing homes). Figure 4.5(b) shows that there is a pronounced variation in this regard. Specifically, Cluster1 and Cluster2 had larger magnitude of patients discharged to long-stay care facilities, compared to Cluster3. This could indicate that more elderly patients are more likely to be discharged to nursing homes or assisted-living residence for example.



(a) Patient age in clusters. (b) Percentages of discharge destinations.

Figure 4.5: The variations of age and discharge destinations in clusters.

Other relevant issues of concern were also examined, including gender and fracture types. Figure 4.6 gives the proportions of male and female patients in clusters. It is clear that the percentages of female patients were much higher in all clusters. Figure 4.7 shows the percentages of fracture types, whereas the proportions of fracture types remained largely similar in the three clusters.

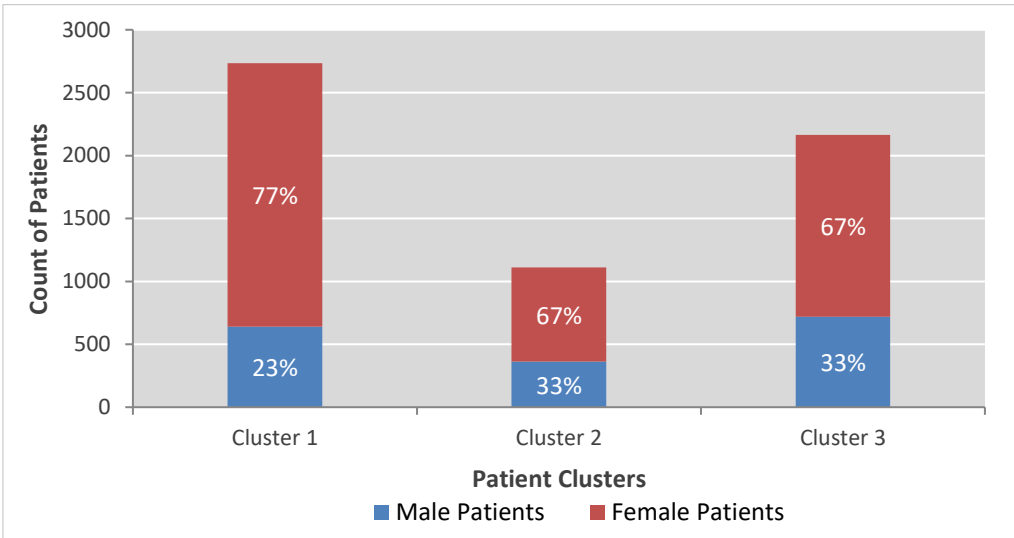


Figure 4.6: Distribution of male and female patients within the three clusters.

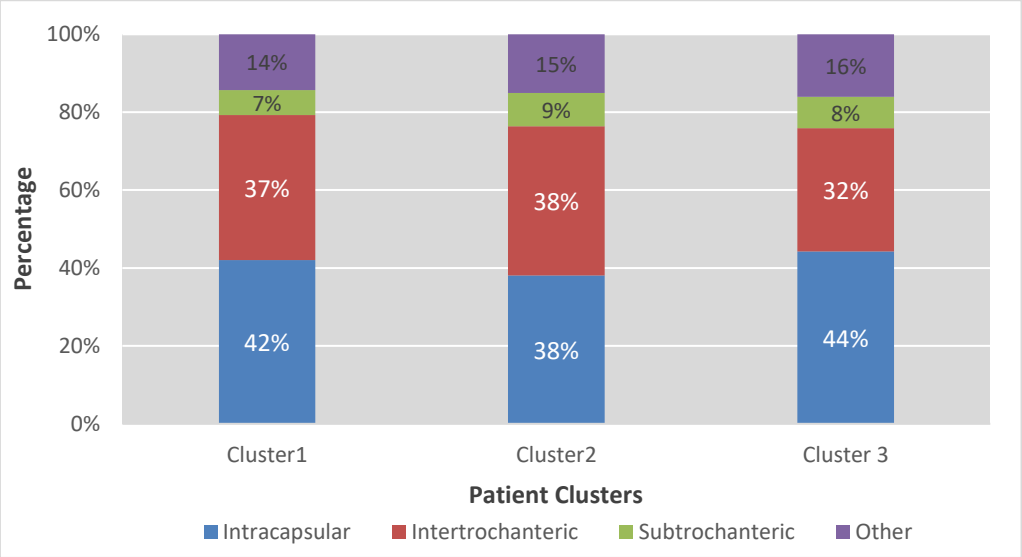


Figure 4.7: Percentages of fracture types in clusters.

4.3 Association Rule Mining

This part describes the second stage of knowledge discovery through the use of association rule mining. The following sections explain the data pre-processing procedures, and highlight the rules discovered.

Objectives

The intention here was to explore whether care outcomes could be associated with a combination of patient attributes and/or care-related factors. Association rules can describe such combinations in a self-explanatory form as (If X Then Y), whereas X is a set of antecedents (i.e. patient attributes, or other factors), and Y represents the consequent (i.e. care outcome).

To focus the experiments, the study was particularly interested in deriving robust rules associated with the inpatient LOS as a care outcome. For example, patients who spend longer periods prior to surgery may be more prone to experience longer LOS. Section 4.2 discussed earlier the vital importance of LOS within hip fracture care in particular, and healthcare in a broader sense.

Rule Mining Algorithm

The study used the Apriori algorithm, widely adopted for rule mining. The Apriori was introduced in the seminal work by Rakesh Agrawal and other fellows from the IBM Almaden Research Center (Agrawal, Imieliński, & Swami, 1993; Agrawal & Srikant, 1994). It has become a standard approach for rule mining tasks. The algorithm seeks to find out frequent itemsets in a dataset of transactions. The typical example is the market-basket analysis, where a set of items tend to be purchased together. However, that basic concept was successfully extended to a broad diversity of domains (e.g. Cooley, Mobasher, & Srivastava, 1997; Hu & Liu, 2004; Srinivasan, 2004).

The Apriori algorithm was cleverly crafted based on the idea of monotonicity of frequent itemsets. The monotonicity here means that an itemset cannot be frequent unless all its sub-items are frequent as well. This means that the algorithm does not have to find the frequency of any set that has a non-frequent subset, and this in turn greatly reduces the number of counts to maintain in the main memory. Figure 4.8 sketches the iterative process of Apriori, while Figure 4.9 gives the algorithm pseudocode.

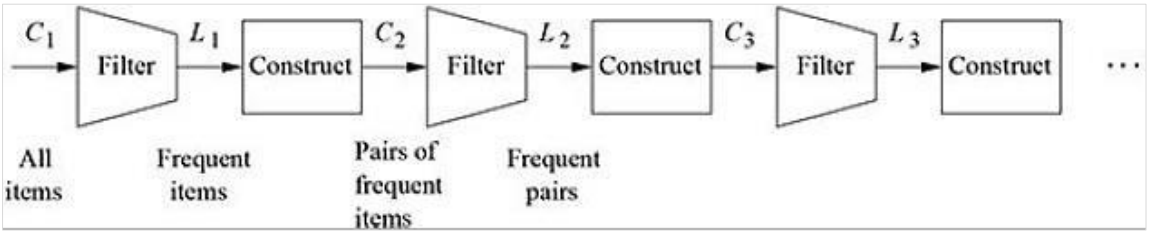


Figure 4.8: The iterative steps of the Apriori algorithm (Leskovec, Rajaraman, & Ullman, 2014).

```
Let k=1
Ck: candidate itemset of size k
Lk: Frequent itemset of size k
L1 = {Frequent Items}
While Lk !=Ø
Ck+1 = Candidate itemsets generated from Lk
For each transaction t
    Increment the count of candidates in Ck+1 that are contained in t
LK+1 = Candidates in CK+1 with the minimum support
End
Return ∪k LK
```

Figure 4.9: Pseudocode of Apriori algorithm.

Discretisation of Numeric Features

The discovery of association rules depends on finding frequent groups of items that exist together in the form of transactions. Typically, rules can be better generalised based on categorical-valued features due to their limited combinatorial possibilities. Therefore, the dataset records needed to be mapped to a transaction-based format suitable for rule mining.

On one hand, the raw dataset included some categorical fields by default (e.g. gender, fracture type). On the other hand, a number of features were naturally continuous-valued including age, TTS and LOS. Those numeric features were transformed into a categorical form through a simple discretisation operation. The features were discretised based on the following criteria:

- **LOS:** The median value of LOS for hip-fracture patients was reported as 12.5 days (NOCA, 2015), and the actual average in the dataset was similarly about ≈ 12.8 days (excluding outliers). Therefore, the LOS values were decided to be classified into two categories as: i)Average (<=13 days), and ii)Above-Average (>13 days).
- **TTS:** The TTS was transformed into three categories: i)24hrs, ii)48hrs, and iii)Above-Standard. This categorisation was based on the quality measures defined by (British Orthopaedic Association, 2007), whereas all hip-fracture patients should undergo surgery within 48 hours of admission.
- **Age:** The categories of patient age were solely predicated on our intuition, whereas patients were divided up into two age groups as: i) AgeGroup1 (60≤ Age ≥80), and ii) AgeGroup2 (Age>80). Table 4.5 presents the discretised features, categories, and associated intervals.

Table 4.5: Discretisation of continuous features.

Feature Name	**Categories**		
	Category Name	**Lower Bound**	**Upper Bound**
LOS (days)	Average	>=1	<=13
	Above-Average	>13	<= 60
TTS (hours)	24hrs	>=1	<=24
	48hrs	> 24	<=48
	Above-Standard	>48	<= 240
Age (years)	AgeGroup1	>=60	<= 80
	AgeGroup2	>80	-

Feature Construction from Patient Clustering

Beyond the discretisation of features, the study availed of the clusters discovered earlier to provide further classification of the transactions. This can effectively designate a group of similar variables by a cluster label, which becomes an additional feature. The idea of utilising clustering for feature construction was well-recognised in ML research such as (Guyon, & Elisseeff, 2003). All data pre-processing procedures were conducted using the R language. Table 4.6 presents a snapshot of the transactions.

Table 4.6: Samples of the transactions.

Gender	Age	Fragility	Fracture Type	Diagnosis (ICD-10)	TTS	Cluster	LOS
Male	AgeGroup2	NO	Intracapsular-Displaced	S7201	24hrs	Cluster1	Avg
Female	AgeGroup1	YES	Intracapsular-Displaced	S7201	24hrs	Cluster3	Above-Avg
Female	AgeGroup2	YES	Intertrochanteric	S7200	24hrs	Cluster1	Above-Avg

Experimental Results

The extraction of rules was conducted over a sequence of experiments. Each rule was associated with measures that reflected the degree of interestingness. Specifically, the support and confidence represented the statistical significance and certainty of rules respectively. The support and confidence can be evaluated as below:

$$Support(X \rightarrow Y) = \frac{N_{X \cup Y}}{N} \qquad Confidence(X \rightarrow Y) = \frac{N_{X \cup Y}}{N_X}$$

Where N is the number of transactions,
N_X is the number of transactions covering X,
$N_{X \cup Y}$ is the number of transactions covering both X and Y.

The experiments were designed using a combination of support/confidence values. The support and confidence were initially set as 10% and 70% respectively. Eventually, 19 rules were extracted at 10% support and 80% confidence. Table 4.7 summarises the experiments, while Table 4.8 lists the discovered rules. The experiments were implemented using the R language along with the *arules* package developed by (Hahsler, Grün, & Hornik, 2005).

Table 4.7: Summary of rule mining experiments.

Experiment	Support	Confidence	Count of Rules
#1	0.1	0.7	62
#2		0.75	54
#3		0.80	19

Table 4.8: The discovered rules.

Rule	Support	Confidence
{Fragility=NO, Cluster=Cluster2} → {LOS=Above-Avg}	0.151	0.998
{Gender=Female, Cluster=Cluster2} → {LOS=Above-Avg}	0.124	0.997
{Fragility=NO, Age= AgeGroup2, Cluster=Cluster2} → {LOS=Above-Avg}	0.110	0.997
{Cluster=Cluster2} → {LOS=Above-Avg}	0.184	0.996
{Age= AgeGroup2, Cluster=Cluster2} → {LOS=Above-Avg}	0.134	0.996
{Fragility=NO, TTS=24hrs, Cluster=Cluster3} → {LOS=Avg}	0.160	0.828
{Fragility=NO, Age=AgeGroup1, TTS=24hrs, Cluster=Cluster3} → {LOS=Avg}	0.160	0.828
{Gender=Female, Fragility=NO, TTS=24hrs, Cluster=Cluster3} → {LOS=Avg}	0.107	0.828
{Gender=Female, Fragility=NO, Age=AgeGroup1, TTS=24hrs, Cluster=Cluster3}→{LOS=Avg}	0.107	0.828
{Diag=S7200, Cluster=Cluster3}→{LOS=Avg}	0.110	0.824
{Diag=S7200, Age=AgeGroup1, Cluster=Cluster3}→{LOS=Avg}	0.110	0.824
{Gender=Female, TTS=24hrs, Cluster=Cluster3}→{LOS=Avg}	0.137	0.824
{Gender=Female, Age=AgeGroup1, TTS=24hrs, Cluster=Cluster3}→{LOS=Avg}	0.137	0.824
{TTS=24hrs, Cluster=Cluster3}→{LOS=Avg}	0.197	0.820
{Age=AgeGroup1, TTS=24hrs, Cluster=Cluster3}→{LOS=Avg}	0.197	0.820
{Gender=Female, Fragility=NO, Cluster=Cluster3}→{LOS=Avg}	0.151	0.809
{Gender=Female, Fragility=NO, Age=AgeGroup1, Cluster=Cluster3}→{LOS=Avg}	0.151	0.809
{Fragility=NO, Cluster=Cluster3}→{LOS=Avg}	0.234	0.803
{Fragility=NO, Age=AgeGroup1, Cluster=Cluster3}→{LOS=Avg}	0.234	0.803

Reflections on Discovered Rules

The association rules were inspected in order to highlight factors that can have an impact on the inpatient LOS. Obviously, all the rules related to Cluster2 patients were associated with longer LOS (i.e. Above-Avg). It is noteworthy that Cluster2 patients were those who experienced longer periods of TTS. Also, Cluster2 and Cluster1 were observed to include the most elderly patients of the dataset. However, it turned out from the rules that TTS may have more influence on LOS compared to patient's age. Further, the rules indicated that the fragility history of patients may have no considerable impact on the LOS.

The significance of TTS appeared clearly again within the rules associated with average-LOS. Specifically, about 60% of the average-LOS rules were associated with TTS within 24 hours. Further, all of those rules were associated with patients belonging to Cluster3, which included less elderly patients aged <=80 (i.e. AgeGroup1). This interpretation of rules largely conformed with the cluster analysis.

The following plot visually illustrates items and rules as vertices connecting them through a graph-based representation (see Figure 4.10). The graph-based techniques attempt to visualise association rules using vertices and edges where vertices typically represent items and edges indicate relationship in rules. Interest measures can be added to the plot as labels on the edges or by color or width of the arrows displaying the edges. In general, graph-based visualisation offers a clear representation of rules.

In Figure 4.10, the node size represents the support value, while the colour corresponds to another measure of rule interestingness, known as the lift. The lift indicates the robustness of a rule with the random occurrence of the antecedent and the consequent. The larger the lift value, the more significant the association between antecedents and consequents. The visualisation was produced by the the *arulesViz* R-package by (Hahsler & Chelluboina, 2011).

Figure 4.10: Visualisation of discovered rules.

4.4 Limitations

There are some limitations of that need to be highlighted as follows:

- The clusters of patients were suggested based on a mere data-driven standpoint. The use of auxiliary domain knowledge could group patients differently.
- All observations and insights delivered by this chapter should be carefully considered as potential correlations limited by the dataset size, and number of variables explored.
- Only public acute hospitals were included in the IHFD data repository.
- The dataset records did not evenly represent the 9 CHO regions of the Irish healthcare system.

4.5 Summary

The main objective in this chapter was to learn and present data-driven insights from the Irish Hip Fracture Database. The knowledge discovery process included the use of data clustering and rule mining. Initially, data clustering was utilised to discover inherent patient clusters in the dataset. The discovered clusters served as a robust basis for exploring potential correlations among patient profiles, care-related factors, and outcomes. It was indicated that the group of patients who experienced longer periods of TTS tended to have remarkably longer inpatient LOS. Also, the clusters of relatively older patients consistently had the highest proportion of patients discharged to long-stay care facilities such as nursing homes.

Furthermore, the rule mining-based analysis put the cluster insights on firmer ground. The association rules marked the impact of TTS on the LOS for hip-fracture patients. Specifically, all discovered rules with LOS longer than average were also associated with TTS above standard (i.e. > 48 hours). On the other hand, patients who went through surgery within 24 hours from admission were observed to spend average LOS period (i.e. <=13 days). From a medical standpoint, the data-driven insights delivered in this study largely conform to the quality standards suggested by (British Orthopaedic Association, 2007).

The knowledge gained will be utilised in the following chapters within building simulation models of the patient's journey. In a broader context, the data-driven insights can be employed for different purposes including medical diagnosis and prediction, care planning, or improvement of care.

Chapter 5

Systems Modelling Aided by Machine Learning: Towards More Data-Driven Feedback Loops

5.1 Introduction

Systems modelling can be contemplated as the science of complexity. Meadows (2008) emphasised that systems inherently exhibit adaptive, dynamic, goal-seeking, self-preserving, and sometimes evolutionary behaviour. This can make a system more than simply the sum of its parts. Therefore, the process of learning about systems of high complexity can be a challenging task.

From a different perspective, systems can now be dealing with extraordinary amounts of data (i.e. Big Data). It should be taken into account that systems involved with such Big Data scenarios place further burdens on the modelling process, which can go beyond human capabilities in many aspects. For instance, the knowledge describing a system may need to be learned or extracted from huge amounts of data, which may be accumulating with a high velocity as well. Viewed this way, the complexity of systems can also be interpreted in terms of the complexity of data encompassing the system knowledge.

In this respect, this chapter endeavours to investigate the feasibility of utilising ML techniques in order to understand, and learn about the underlying structure or behaviour of systems. Our fundamental view is that system models can be developed based on knowledge learned by ML models in tandem with mental models-driven knowledge. The study specifically focused on the SD approach, as a widely used method for systems modelling.

Developed by Jay Forrester in the 1950s, the SD approach aimed to introduce a set of tools that enabled to understand the structure and dynamics of complex systems. Over decades, SD has successfully established a large community in academia and industry, and continued to become an instrumental artefact for finding appropriate strategies and policies by elucidating the dynamic behaviour of systems.

The chapter can be viewed as organised into two main parts as follows. The first part initiates a discussion regarding the process of learning about systems from the perspective of feedback loops. That discussion was aimed to serve as an opening to the rationale behind the proposed approach. Afterwards, the second part, starting from Section 6, provides a more practical standpoint based on a use case in relation to healthcare. The use case was mainly adopted in order to present realistic scenarios, where systems modelling can be supported by data-driven insights gained through ML. Specifically, we availed of the data-driven insights learned by the clustering model as presented in Chapter 4.

5.2 The Feedback Loop Concept

To set a context, we initially intended to drive the discussion through the concept of feedback loop. The feedback concept has been an essential component for the SD approach, and systems modelling in general. This section briefly reviews how that concept was perceived within the context of systems modelling, and ML as well.

The Feedback Loop in System Dynamics

The feedback loop concept has its roots in different disciplines that go back further beyond the development of SD. Richardson (1983) provided a comprehensive historical review of that concept, and how it was endorsed in a number of sciences. The review argued that the feedback concept has evolved and matured as a blend of ideas from different sources of subjects including: i) Engineering, ii) Biology, iii) Mathematical models of biological and social systems, and iv) Social sciences. Richardson presented interesting examples of feedback loops in engineering that dated back to 250 B.C.

From the perspective of the SD approach, the concept of feedback has been embraced as one of the core ideas within the modelling process. Forrester (1960; 1961; 1964) consistently asserted the importance of feedback loops within systems, and that all decisions are developed within the context of feedback loops. Forrester (1968) described the feedback loop as a closed path connecting in sequence a decision controlling an action, a state of the system, and information about the state, which returns to the decision-making point. Figure 5.1 sketches the feedback loop in its simplest form.

Further articulations of the feedback loop were compiled by (Sterman, 1994). Sterman discussed issues pertaining to feedback loops, and illustrated by examples its significance in a broader context. Equally important, Sterman's work endorsed the inevitable influence of

mental models while perceiving information from feedback loops, and making decisions. Figure 5.2 (a) re-portrays the feedback loop in view of existing premises derived from mental models. Furthermore, Figure 5.2 (b) demonstrates that feedback from the real world can also cause changes in our mental models.

A more complex adaptation of Sterman's view was developed by (Sadsad, & McDonnell, 2014), as sketched in Figure 5.3. Their view included the process of gathering of knowledge, evidence and theory to be synthesised and analysed, along with the communication and interpretation of results to inform real-world decisions. This perspective can adequately depict the multi-loop and nonlinearity of real-world systems involved in many simulation studies.

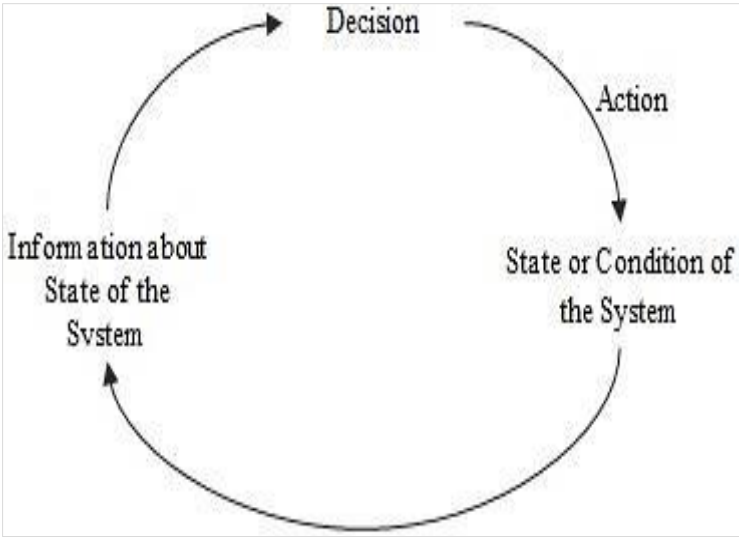
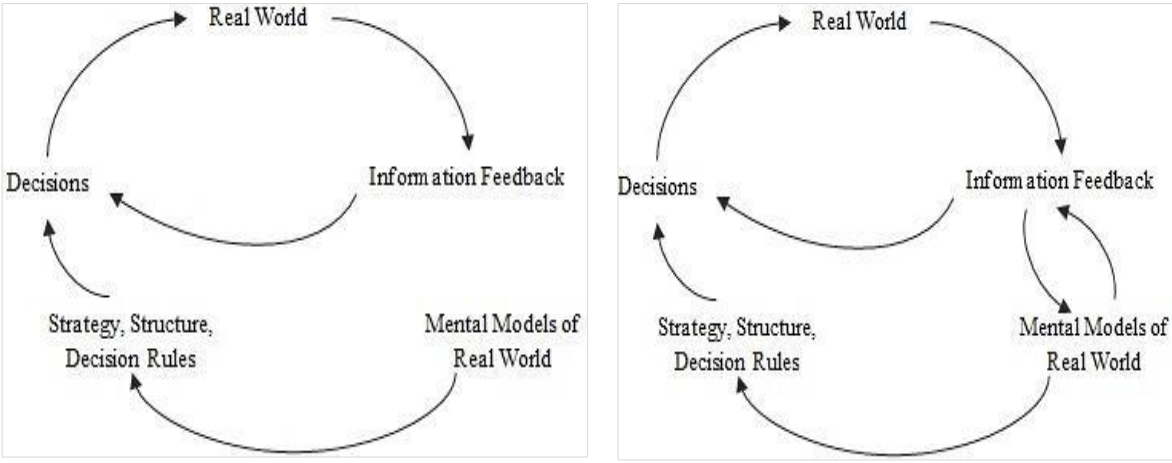


Figure 5.1: The basic feedback loop concept.



(a) **(b)**

Figure 5.2: The role of mental models in the feedback loop.

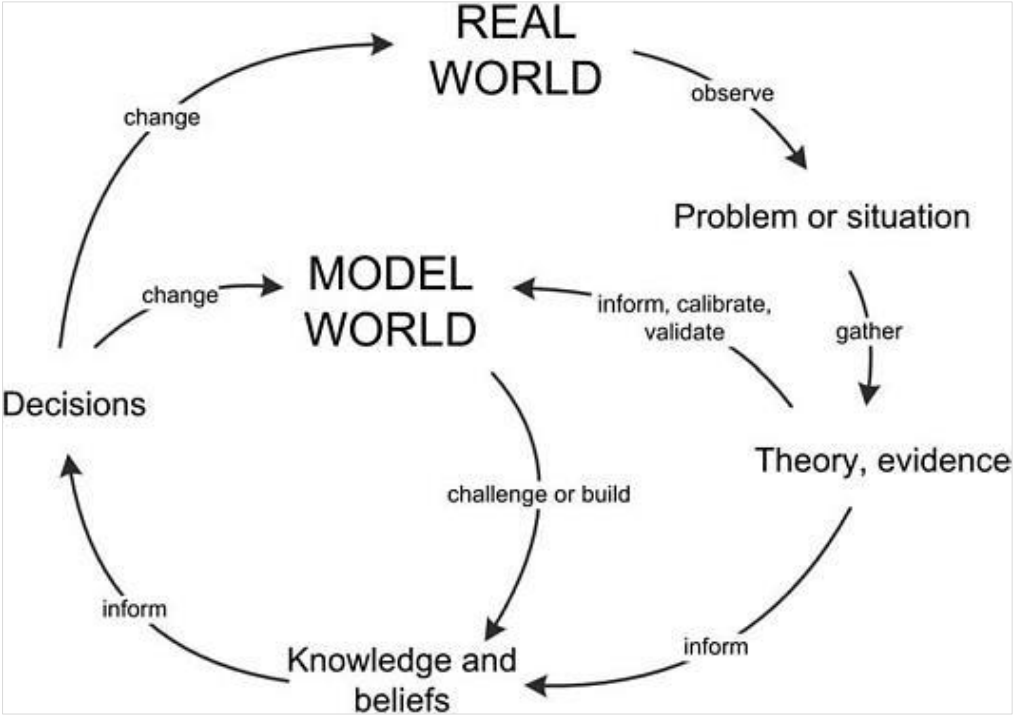


Figure 5.3: A more complex version of feedback Loop (Sadsad, & McDonnell, 2014).

Limitations of Mental Models

In his landmark textbook (The Fifth Discipline), Senge (1990) described mental models as the deeply ingrained assumptions, generalisations, or even images that affect how we understand the world, and how we take actions. From the very beginnings of SD development, Forrester (1961; 1971) highlighted the unavoidable limitations of our mental models. Forrester emphasised that the mental model is fuzzy, incomplete, and imprecisely stated. However, Sterman (1994) argued that most people do not appreciate the ubiquity and invisibility of mental models. In this regard, Sterman identified a set of mental models-related barriers to learning feedback as follows:

- Misperceptions of feedback.
- Flawed cognitive maps of causal relations.
- Erroneous inferences about dynamics.
- Unscientific reasoning.
- Judgmental errors and biases.
- Defensive routines and interpersonal impediments to learning.

It can be understood that the above-mentioned barriers are largely attributed to the nature of human-based reasoning, which could be predicated on biased perception of information. The initial view of the study was that more machine-oriented assistive methods (e.g. ML) may constitute a key factor to mitigate such limitations.

The Feedback Loop in Machine Learning

This section aims to review how the feedback concept was also recognised within the context of ML. A number of approaches were developed to organise the process of data mining, or knowledge discovery. The widely used approaches are: i) CRISP-DM (Cross-Industry Standard Process for Data Mining) (Shearer, 2000), ii) SEMMA (Sample, Explore, Modify, Model, and Assess,) and iii) Knowledge Discovery in Databases (KDD). Detailed explanations of those approaches would go beyond the scope of this study, since this section is merely interested in highlighting the existence of feedback loops in ML context. However, the study may suggest other sources such as (Azevedo, & Santos, 2008) and (Oslon, & Delen, 2008) for understanding the different aspects of every approach.

Though not being mentioned explicitly, the notion of feedback can be inferred within the three approaches. Starting with the CRISP-DM (Figure 5.4), the surrounding circular loop signifies the need for a continuous feedback through data mining procedures. Similarly, the KDD approach (Figure 5.5) utilises inter-linked arrows representing the possibility for making a U-turn to update the current understating, or methods used. In the SEMMA approach (Figure 5.6), the large arrow again considers possible feedback within every procedure.

However, feedback-based knowledge here can be learned and tested more robustly. This can be attributed to the highly data-driven nature of ML, which depends on clear-cut evaluation measures for scoring model accuracy. For instance, the accuracy of a classifier or a regression model can be clearly expressed in terms of quantitative measures (e.g. precision-recall). Therefore, it can be quite straightforward to compare the accuracy of many competitive ML models, or to set a threshold level of accuracy. In contrast, such measures hardly exist within the context of modelling and simulation. That is why the mental models of experts, modellers, or simulationists could have a higher influence within the process of establishing or updating knowledge learned through feedback loops.

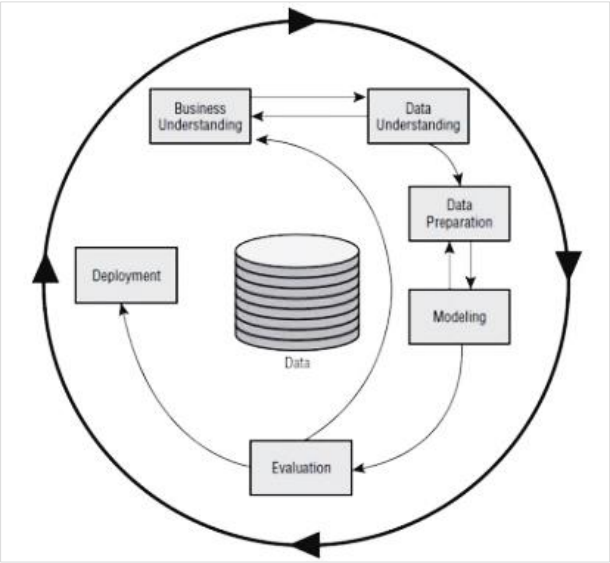


Figure 5.4: The CRISP-DM.

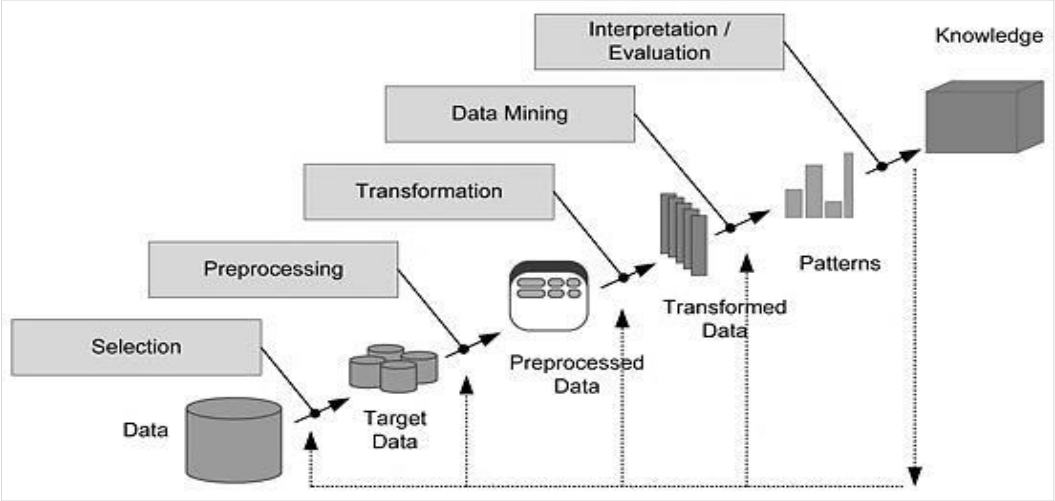


Figure 5.5: The KDD process.

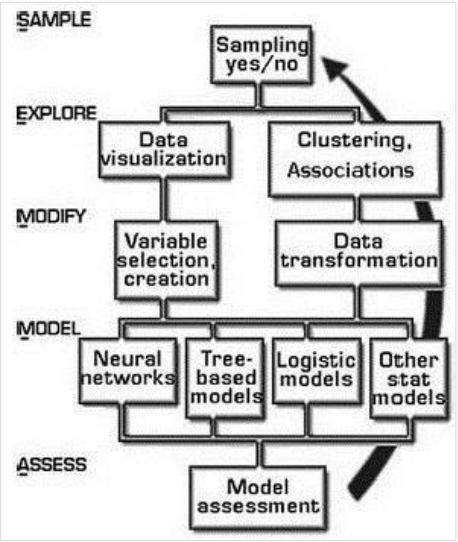


Figure 5.6: The SEMMA approach.

5.3 The Prospective Role of Machine Learning in Systems Modelling

> *"Even though the assumptions of a model may not literally be exact and complete representation of reality, if they are realistic enough for the purpose of our analysis, we may be able to draw conclusions which can be shown to apply to the world."*, (Cohen, & Cyert, 1965).

The 'realism' of simulation models has been an active point for discussion in systems modelling research. As quoted above, a simulation model is not expected to be a complete representation of reality to deem useful. A simulation model should largely attempt to represent a sufficient level of reality, which can be acceptable regarding the questions of interest. The established argument is that all models, including simulation models, are considered invalid regarding complete reality (e.g. Stanislaw, 1986; Shreckengost, 1985).

In this respect, the integration of Simulation Modelling with ML may help simulations attain a higher level of model realism. The initial view was that ML can be employed as an assistive tool to help reduce the epistemic uncertainty (Oberkampf et al., 2002) underlying simulation models. That kind of uncertainty is largely attributed to the subjective interpretation of system knowledge by modellers, simulationists, or subject matter experts. In other words, if simulation models could be supported by predictive models trained to make predictions on the actual system's behaviour, this could in turn lead to a relatively lower degree of uncertainty.

From a more practical standpoint, ML can be utilised in order to predict the behaviour of system variables that may not be feasible to express analytically. For example, (Zhong, Cai, Luo, & Zhao, 2016) trained ML models within a use case for crowd modelling and simulation. The ML models were used to learn and predict the flow of crowds. Likewise, unsupervised ML techniques (e.g. clustering) can be used to learn about key structural characteristics of systems, especially in case of Big Data scenarios.

5.4 Hybrid Modelling: Mental Models Aided by Machine Learning Models

"We are flooding people with information. We need to feed it through a processor. A human must turn information into intelligence or knowledge" – Grace Hopper.

In the sense of that quote by Grace Hopper, one of the early computing pioneers, the study approach mainly aimed to support mental models with data-driven knowledge learned by ML. The key idea hinges on the premise that mental models can be assisted by ML models trained to make predictions on a particular aspect of the system's structure, or behaviour being modelled. It is assumed that changes in the sates or conditions of a system can be inferred, at least partially, through ML predictions.

As illustrated in Figure 4.7, new data (i.e. feedback) can be generated by new states of the system of interest. Based on data-driven feedback, ML models can be trained to predict the future behaviour of the system. Moreover, ML models can be continuously re-fitted to echo feedback loops, and reflect new system's conditions. In this manner, behavioural changes can be learned based on ML models in tandem with mental models. The under-consideration argument is that relatively unbiased, or less biased, data-driven predictions can help improve the understanding of systems, and in turn more accurate decisions can be made.

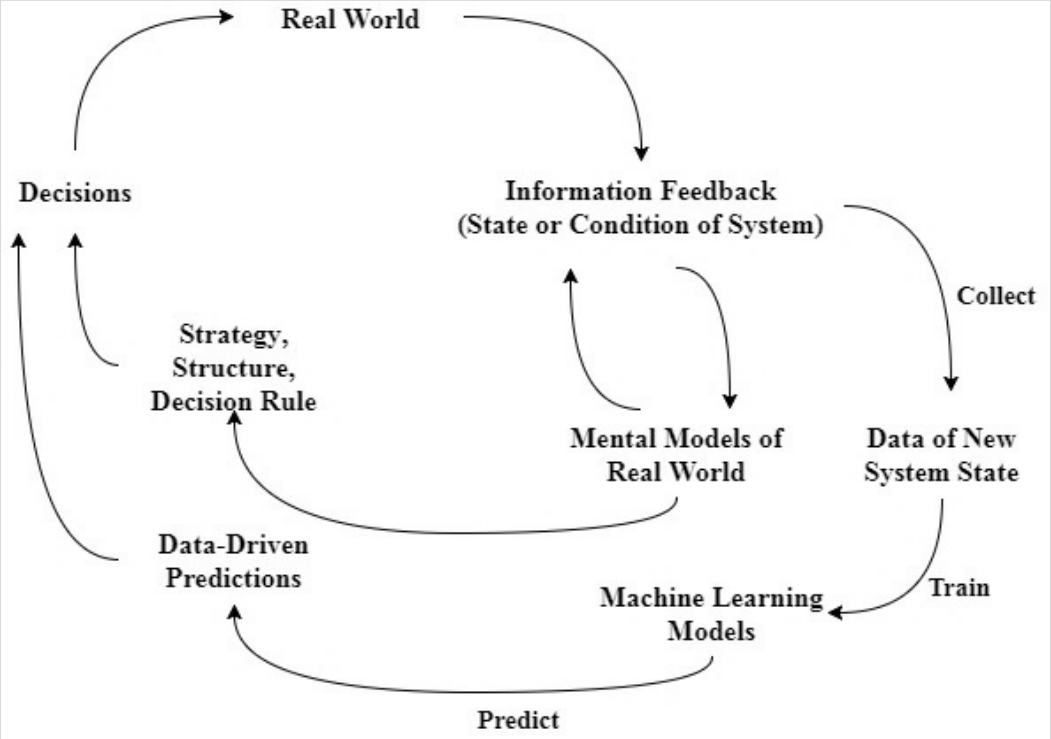


Figure 5.7: Approach overview.

5.5 Use Case: Modelling Flow of Elderly Patients

For the purpose of demonstrating the applicability of our approach, a case study was developed in relation to healthcare. The following sections elaborate the case setting, and the development of ML and SD models. The dataset used was already described in detail at Chapter 2. The main goal of the use case was to provide a practical scenario where SD models can be designed or adjusted in accordance with new system conditions learned by the aid of ML.

Case Description

Within the healthcare context, the use case particularly focused on the care of hip-fracture patients in Ireland. Hip fractures are a major cause of injuries and morbidity among the elderly. As acknowledged by numerous studies (e.g. Melton, 1996), hip fractures were observed to be exponentially increasing with age, despite the existence of rate variability from country to another. Further, the care of hip fractures has a considerable importance, whereas the Ireland's Health Service Executive (HSE) identified hip fractures as one of the most serious injuries resulting in lengthy hospital admissions and high costs (HSE, 2008).

The typical patient's journey is described as follows. Initially, a patient is usually received at the ED. Within 48 hours, the primary surgery should be performed after admission to an orthopaedic ward. Subsequently, the patient can possibly undergo various assessments based on falls history and fragility. Eventually, the discharge destination is mainly decided as: i) Home, or ii) Long-stay care facility (e.g. nursing home).

Purpose of the Model

The study aimed to develop an SD model that can depict the flow of elderly hip-fracture patients from admission to discharge. The model can be utilised to understand and estimate the potential demand of care against the capacity of healthcare facilities. Specifically, the model focused on the utilisation of healthcare facilities in terms of: i) Inpatient length of stay (LOS), and ii) Discharge destinations. To focus the purpose of the model, the questions of interest are stated as below:

1. What is the expected consumption of hospital resources with regard to the inpatient LOS?
2. What is the expected proportion of elderly patients discharged to home, or long-stay care facilities?

Selection of Machine Learning Technique

As per the questions posed in Chapter 1, it was initially aimed to investigate the possible ML techniques that can be used to assist the process of problem conceptualisation. In this regard, unsupervised ML techniques (e.g. clustering) present adequately for perceiving the system's structure or behaviour as follows.

The SD models typically deal with aggregate entities (e.g. population of patients), and not individual entities or agents. Those aggregate entities are represented as stocks, which characterise the state of the system and generate the information upon which decisions and actions are made (Sterman, 2000). In a relatively similar manner, clustering seeks to realise the segmentation of a heterogeneous population into a number of more homogeneous subgroups (Aldenderfer, & Blashfield, 1984). Viewed this way, the suggested homogeneous groups (i.e. clusters) may correspond to particular stocks in the SD model. In addition, clustering is an effective method for exploring potential underlying structures in the system without making any prior assumptions that might be biased.

5.6 System Dynamics Modelling

Initial SD Model

The initial model provided a bird's-eye view of the care scheme of hip fractures with respect to the questions of interest. The model focused on capturing the dynamic behaviour in relation to the continuous growth of ageing, and the consequent implications on the incidence of hip fractures among the elderly. The main actors within the model were defined as follows: i) Elderly patients, ii) Acute hospital, and iii) Discharge destinations including home or long-stay care facilities.

Two different inflow rates were set for male and female patients, which were defined by (Dodds, Codd, Looney, & Mulhall, 2009). The model included a single reinforcing loop implied by the elderly patients with fragility history, who are susceptible to re-sustain hip fractures, or fall-related injuries at least. At this stage, the model did not consider the different characteristics of patients learned by the ML clustering or rule mining experiments. Figure 5.8 illustrates the initial SD model.

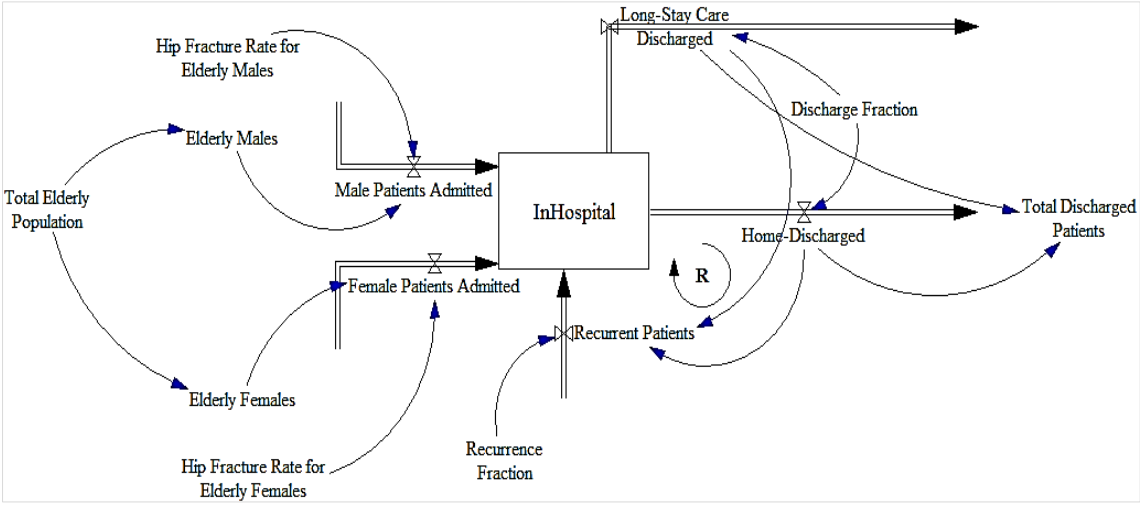


Figure 5.8: Initial SD model.

Cluster-Based SD Model

The SD model was re-designed in light of the knowledge learned by clustering experiments in Chapter 4. In particular, the model was disaggregated into 3 different stocks representing the discovered clusters of patients. Furthermore, the model behaviour was mainly set based on the cluster analysis. For instance, the first and second clusters were considered to undergo the same TTS delay (i.e. TimeToSurgery1), while the third cluster was set a different delay (i.e. TimeToSurgery2). Likewise, each cluster was associated with a specific fraction related to discharge destinations (i.e. home, or nursing home).

Equally important, the inflow of elderly patients was structured based on the age groups within clusters. In particular, both of the first and third clusters were modelled to contain more elderly patients (i.e. aged 80-100), while the second cluster was associated with less elderly patients (i.e. aged 60-80). This reflected the age distribution within the clusters, as explained previously in Chapter 4. For the purpose of simplicity, the model did not include the case of recurrent patients, which caused a reinforcing loop in the initial model. Figure 5.9 sketches the cluster-based model.

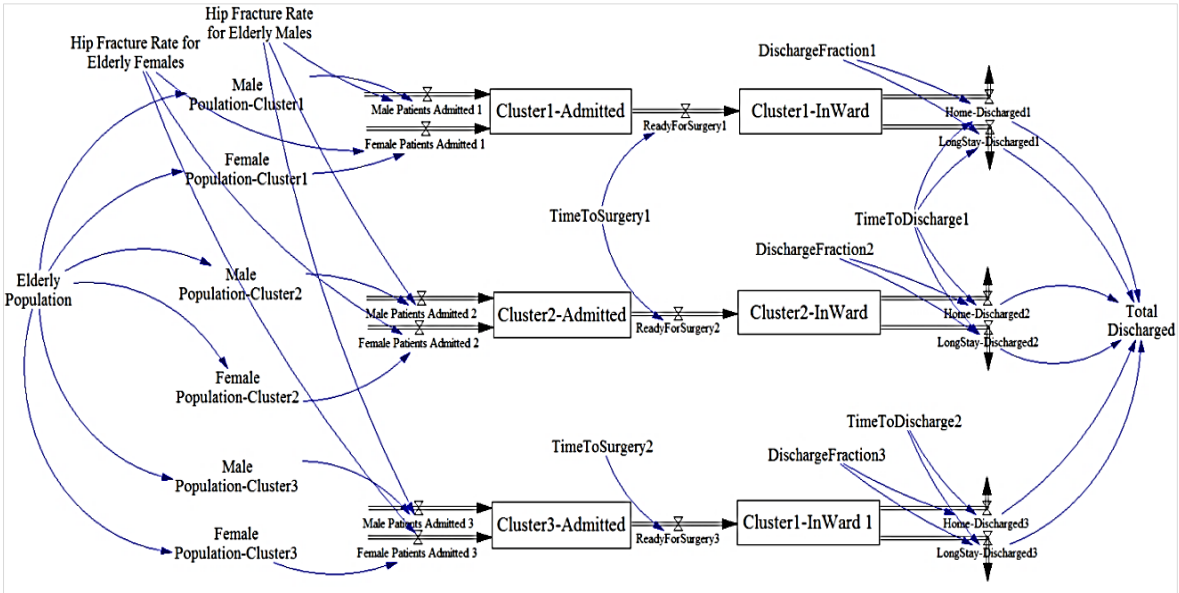


Figure 5.9: The cluster-based SD model.

Simulating Data-Driven Feedback

In order to demonstrate the effect of data-driven feedback learned by ML, the study applied a hypothetical scenario of care improvement. The scenario was intended to simulate a hypothetical change in the system behaviour as follows. It was assumed that a new policy was introduced starting from the year 2014 towards improving the patient's journey. The new policy aimed to maintain the hip-fracture care standards by keeping the TTA and TTS within 4 hours and 48 hours respectively. In accordance with that new policy, the average inpatient LOS was assumed to decrease by 20% and 30% in 2014 and 2015 respectively. Further, the proportion of patients discharged to long-stay residential care was assumed to decrease by 5% and 10% in 2014 and 2015 respectively. In order to reflect the new policy, the patient records of the years 2014 and 2015 were synthetically altered. For instance, the new LOS values was reduced by 20% for patients discharged in 2014.

Subsequently, the clustering model was re-constructed in view of such policy changes. The new clusters are demonstrated in Figure 5.10. It turned out that the new policy led to fewer clusters of patients. Specifically, the finest separation of clusters was realised when K=2. The new clusters were re-explored with respect to the LOS, TTS, and patient age as plotted in Figure 5.11. Based on the new patient clusters, the SD model design was modified. The updated SD model corresponded to knowledge updates learned by the ML clustering model. Figure 5.12 sketches the updated SD model.

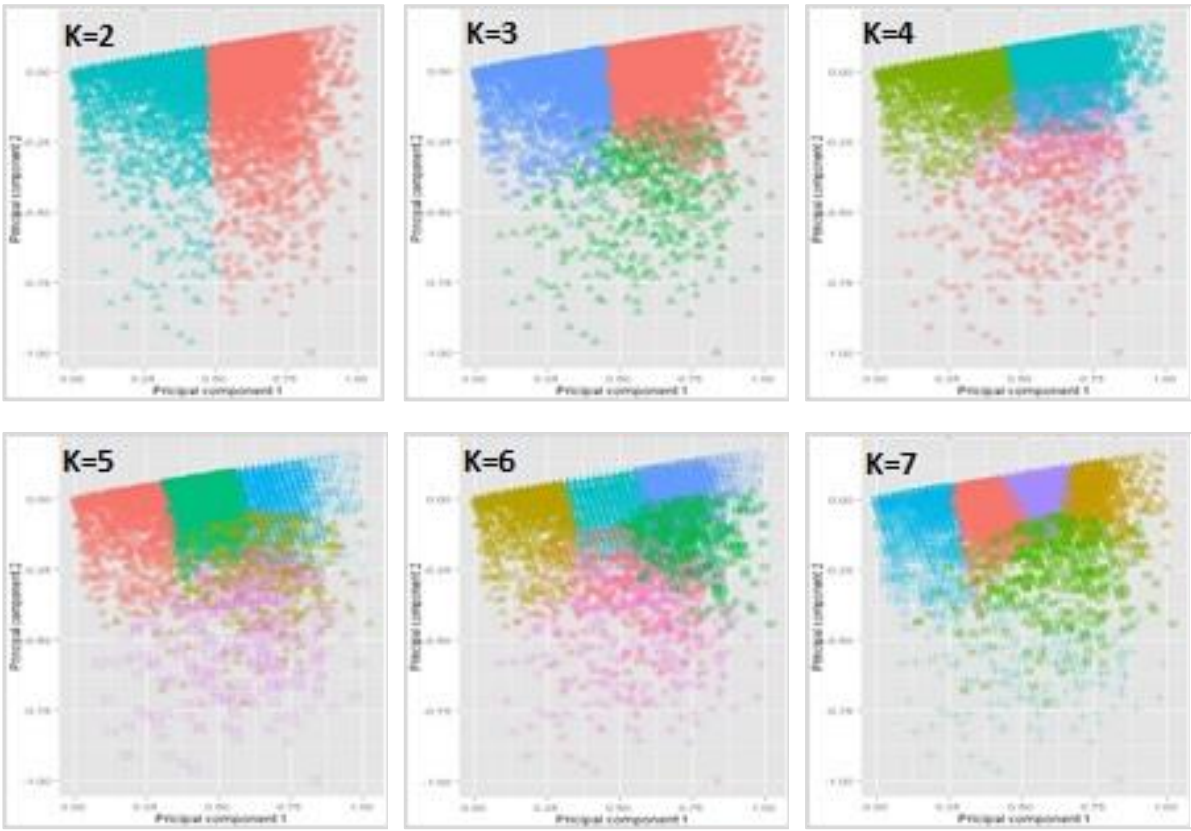
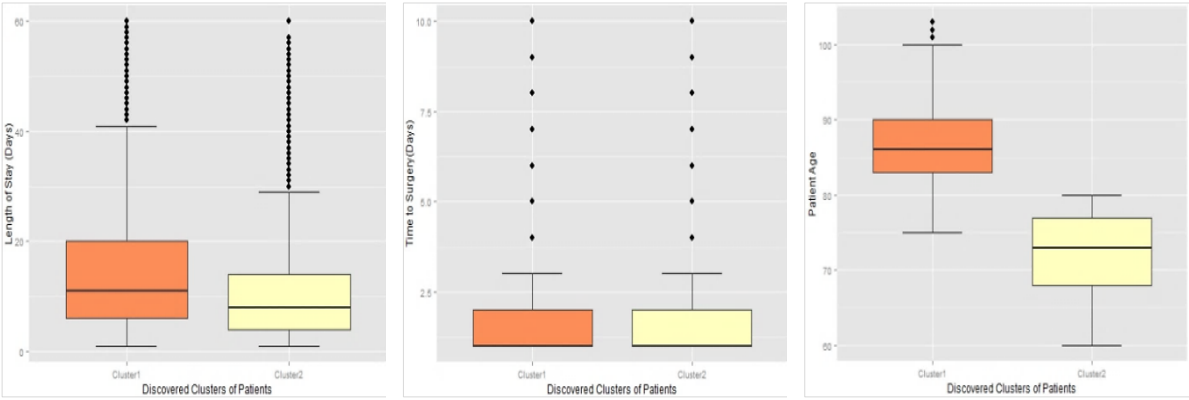


Figure 5.10: Visualisation of clustering experiments after applying the new care policy.



(a) LOS. (b) TTS. (c) Patient age.

Figure 5.11: The variation of the LOS, TTS, and age variables in the new clusters.

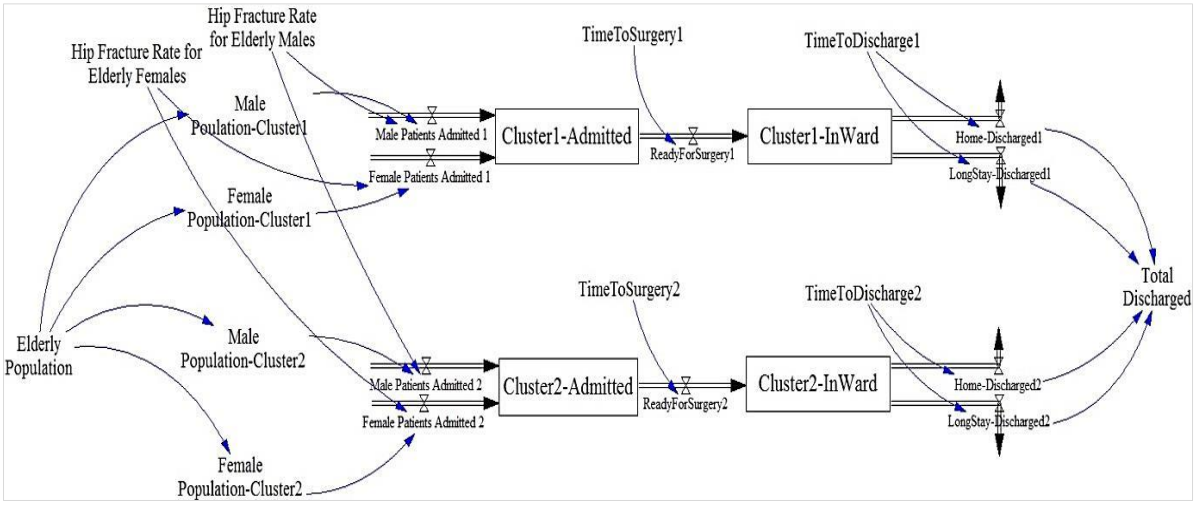


Figure 5.12: The updated cluster-based model.

5.7 Discussion

It is believed that the developed scenario largely addressed the motivational questions listed in Chapter 1. First, the clustering model was employed effectively for the purpose of understanding the system structure, where the SD model stocks actually represented the three discovered clusters of patients. Moreover, the variations within clusters in terms of patient characteristics (e.g. age), or care-related factors (e.g. TTS) assisted with shaping the model behaviour. Furthermore, it can be argued that the SD model was constructed with an established confidence based on the clustering model. The well-validated quality of clusters along with the compelling visualisations could support the rationale behind the SD model design in terms of structure and behaviour as well. Thus, the use of ML could have led to lowering the epistemic uncertainty usually attributed to the subjective interpretation of system knowledge by modellers, or simulationists, as explained by (Oberkampf, 2002).

In our case, the clustering model played an appropriate role to explore possible systemic structures based on a pure data-driven standpoint. However, other ML techniques may be more appropriate within different situations, or other simulation approaches.

5.8 Limitations

A set of limitations are acknowledged as follows. The presented use case may not have been the ideal scenario to demonstrate the potentials of integrating simulation modelling and ML. It is conceived that a typical Big Data scenario can better present the benefits of that integration.

Another relevant issue of concern, the patient clustering was based on a mere data-driven standpoint. Adding a clinical perspective (e.g. diagnosis, procedures) may have an impact on the grouping of patients.

5.9 Summary

Generally, the chapter lent support to the discussion of why and how ML can support the practice of modelling and simulation. It was demonstrated that the integration of mental models with data-driven insights learned by ML models can yield potential benefits for the practice of modelling and simulation. One potential benefit is lowering the bias of mental models, which can in turn increase the confidence in simulation models.

From a practical standpoint, it was attempted to practically demonstrate how ML can assist the process of problem conceptualisation. Based on a use case in relation to healthcare, it was aimed to provide a pragmatic perspective for integrating SD models with data-driven insights learned by ML models. Through a practical scenario, the study utilised utilised ML clustering in order to learn about the system's structure and behaviour. Further, it was demonstrated how changes in clusters reflected on the structure and behaviour of the SD model. This was used as a proxy for the idea of data-driven feedback loops.

Chapter 6

Supervised Machine Learning: Predicting Care Outcomes

6.1 Introduction

In previous chapters, unsupervised ML techniques (e.g. clustering and rule mining) were utilised in order to learn about patients and the care journey in general. The data-driven knowledge gained from unsupervised models helped to design the structure and behaviour of simulation models. This chapter starts a new section of using ML through the development of supervised models (e.g. regression, classification). It was intended to avail of such ML models to provide predictions that can allow for guidance to simulation experiments as explained in the next chapters.

The main focus is placed on predicting care outcomes for hip-fracture patients. Care outcomes were described as the end result of care, or a measurable change in the health status or behaviour of patients (Harris, 1991). According to (Nolan, & Mock, 2000), there are four categories of care outcomes including: i) Clinical outcomes, ii) Functional outcomes, iii) Financial outcomes, and iv) Perceptual outcomes. The clinical outcomes mainly represent the patient's response to medical interventions. The functional outcomes measure the improvement of a patient's physical functioning. While financial outcomes are mainly used to assess the efficient usage of resources. The perceptual outcomes might be the most intangible set of outcomes, which can represent the level of patient satisfaction with care received and its providers. The scope of patient outcomes addressed here can be considered to fall within the clinical and financial categories. Specifically, it was aimed to avail of ML in order to predict the following care-related outcomes:

- The inpatient length of stay of elderly patients.
- Discharge destinations of elderly patients.

Abundant studies emphasised the importance of LOS and discharge destination in healthcare. On one hand, the LOS was suggested as a significant measure of patient outcomes, and a valid proxy to measure the consumption of hospital resources (Faddy, & McClean, 1999). From a financial perspective, it was reported that the LOS represents a major segment of the overall cost of hip fracture treatment (Johansen et al., 2013). On the other hand, the early prediction of patient discharge destinations can hold strategic significance with regard to estimating the appropriate capacity of long-stay care facilities (e.g. nursing homes).

The development of ML models included a regression model for predicting the LOS, and a binary classifier for predicting discharge destinations. The discharge destination was predicted as either home or a long-stay care facility. The ML models were developed using the Azure ML Studio. The following sections go through the data pre-processing procedures and model development in detail.

6.2 Data Anomalies

As mentioned previously in Chapter 2, the study acquired a subset from the IHFD repository. The existence of data anomalies is largely unavoidable. This section goes through the anomalies encountered within the dataset prior to ML.

A data anomaly was defined as an observation that appears to be inconsistent with the remainder of the dataset (Hodge, & Austin, 2004), or more generally as any data that is unsuitable for the intended use (Sarsfield, 2009). This section describes data anomalies exposed in the IHFD dataset, and the procedures conducted to deal with them. In our case, data anomalies mainly existed in terms of outliers, and data imbalances.

Outliers

Outliers existed within the inpatient LOS values, which were exceptionally longer than the mean and median values (i.e. 19 and 12.5 days) (NOCA, 2015). The LOS outliers can be observed in Figure 6.1, which plots a histogram of the LOS variable. In order to prevent the odd influence of outliers, we considered only the samples whose LOS were no longer than 60 days. The excluded outliers represented approximately 8% of the overall dataset.

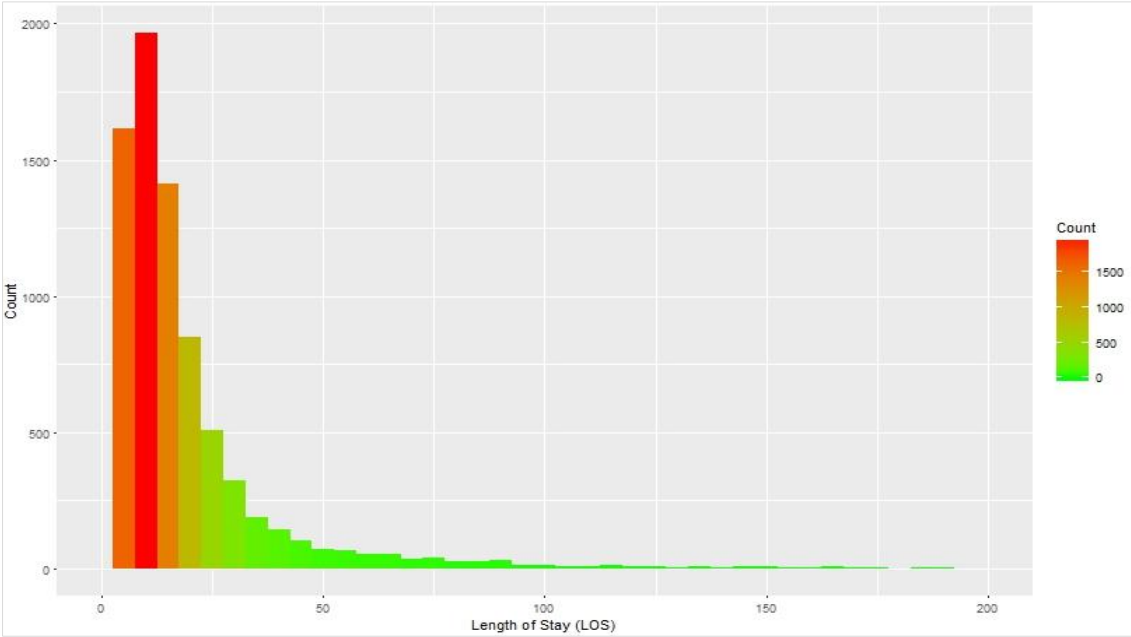


Figure 6.1: Histogram and probability density of the LOS variable. The outliers can be obviously observed when LOS becomes longer than 60 days.

Data Imbalances

Learning with imbalanced datasets was continuously recognised as one of the principal challenges for ML (Yang, & Wu, 2006; Galar et al., 2012). The negative impact of imbalanced data on the prediction accuracy was emphasised by numerous studies such as (Japkowicz & Stephen, 2002), and (Sun, Wong, & Kamel, 2009).

In this study, the training data suffered from imbalanced class distributions, where a particular class pre-dominated the dataset. In particular, imbalanced training samples were pronounced for patients who had LOS longer than 18 days. Likewise, imbalanced samples could be observed for patients who were discharged to home. Moreover, training samples for male patients, and particular age groups were obviously underrepresented.

Two strategies were identified by (Galar et al., 2012) in order to deal with the imbalance problem including: i) Algorithm-level approach, and ii) Data-level approach such as under-sampling or over-sampling. The over-sampling technique purposed by (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) was used in this study. The underrepresented samples were re-sampled at random until they contained sufficient examples. Figure 6.2 and Figure 6.3 show the histograms of LOS and discharge destination respectively, where imbalanced distributions can be clearly observed.

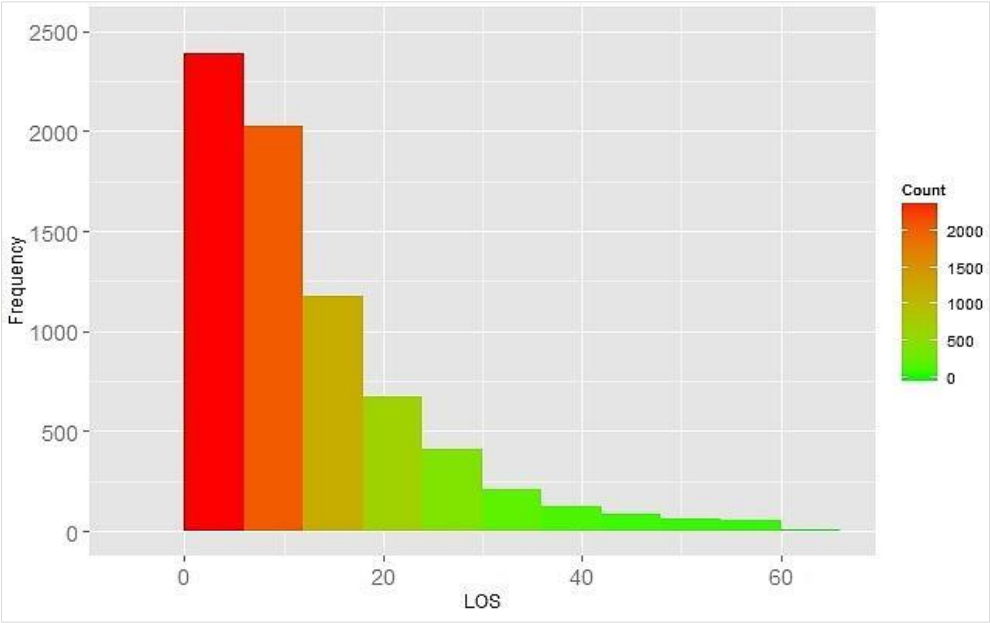


Figure 6.2: Data imbalance (LOS variable).

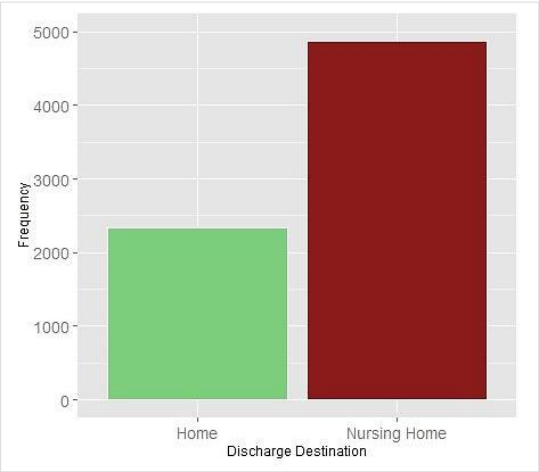


Figure 6.3: Data imbalance (discharge destination variable).

6.3 Feature Selection

Initially, the study explored the variables that can possibly serve as features for training the ML models. Based on intuition, many irrelevant variables were simply excluded (e.g. admission time/discharge time). As mentioned in Chapter 4, the TTA, and TTS represent important quality care-related factors for the hip fracture care scheme. The ML models utilised the TTA and TTS features, which were extracted earlier during the clustering model development.

The most significant features were decided based on the technique of permutation feature importance (Altmann, Toloşi, Sander, & Lengauer, 2010). Table 6.1 lists the variables initially considered as candidate features. Table 6.2 presents the set of features used by both models.

Table 6.1: Variables explored as candidate features.

Variables Explored		
Source Hospital	Admission Type	Discharge Code
Residence Area	Patient Gender	Discharge Status
Admission Source	Hospital Transferred From	Hospital Transferred To
Age	LOS	ICD-10 Diagnosis
Admission Trauma Type	Admission via ED	Fracture Type
Pre-Fracture Mobility	Fragility History	Specialist Falls Assessment
Multi-Rehabilitation Assessment		

Table 6.2: Selected features.

Prediction Model	**Selected Features**
Regression Model-LOS Predictor	Age, Patient Gender , Fracture Type, Hospital Admitted To, ICD-10 Diagnosis, Fragility History, TTS , TTA
Classifier Model-Discharge Destination Predictor	Age, Patient Gender, Fracture Type, Hospital Admitted To, ICD-10 Diagnosis, LOS, Fragility History, TTS

6.4 Learning Algorithm: Random Forests

The Random Forests algorithm (Breiman, 2001) was used for regression and classification as well. A Random Forest consists of a collection of decision trees $\{h(x, \Theta_k), k = 1,...\}$, where the $\{\Theta k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular prediction at input x. The main idea is that for the k_{th} tree, a random vector Θ_k is generated, independently of the past random vectors $\Theta_1,..., \Theta_{k-1}$ but with the same distribution; and a tree is grown using the training set and Θ_k , resulting in a classifier $h(x, \Theta_k)$ where x is an input vector. Figure 6.4 portrays a simple example of a Random Forest composed of 3 decision trees.

Predictions are made through weighted voting for the most likely class, and the trees that have a higher prediction confidence will have a greater weight in the final decision of the ensemble. The aggregation of voting can be done by a simple averaging operation as in the equation below. The developed models consisted of 8 decision trees. Table 6.3 presents the parameters used for training the regression and classifier models.

$$p(c|v) = \frac{1}{T}\sum_{t=1}^{T} p_t(c|v)$$

Where pt(c|v) denotes the posterior distribution obtained by the t-th tree.

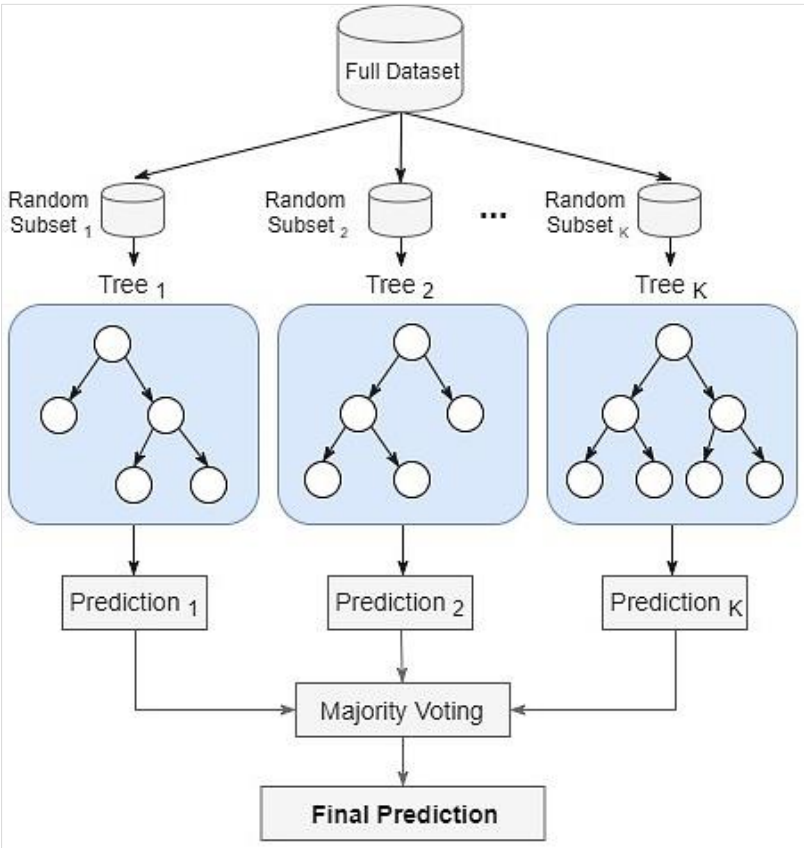


Figure 6.4: Random Forest example: Combining predictions using majority voting.

Table 6.3: Parameters of Random Forests.

No. of Decision Trees	8
Max. Depth of Decision Trees	32
No. of Random Splits per Node	128

6.5 Model Evaluation

The predictive models were tested using a subset from the IHFD dataset. The randomly sampled test data represented approximately 40% of the dataset. The prediction error of each model was estimated by applying 10-fold cross-validation. Table 6.4 and Table 6.5 present evaluation metrics of the regression and classifier models respectively. Further, Figure 6.5 shows the AUC curve of the classifier model.

Table 6.4: Average accuracy based on 10-fold cross-validation (LOS regression model).

Relative Absolute Error	≈ 0.30
Relative Squared Error	≈ 0.17
Coefficient of Determination	≈ 0.83

Table 6.5: Average accuracy based on 10-fold cross-validation (discharge destination classifier).

AUC	≈ 0.87
Accuracy	≈ 80 %
Precision	≈ 81 %
Recall	≈ 79 %
F1 Score	≈ 80 %

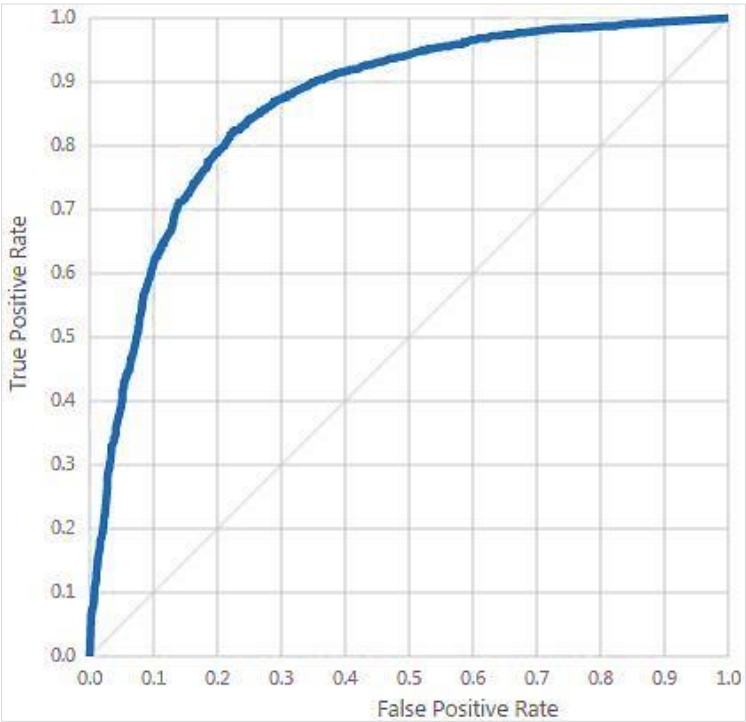


Figure 6.5: Classifier AUC= 0.865 (discharge destination classifier).

The accuracy of various regression and classification techniques was investigated as well. Random Forests proved to outperform other algorithms in our case. Table 6.6 and Table 6.7 provide a comparative analysis with respect to the learning algorithms experimented to develop the regression and classification models.

Table 6.6: Comparison of regression algorithms.

Algorithm	Relative Absolute Error (≈)
Random Forests	0.30
Boosted Decision Tree	0.34
Neural Network	0.55
Linear Regression	0.93

Table 6.7: Comparison of classification algorithms.

Algorithm	Precision (≈)	Recall (≈)	Accuracy (≈)
Random Forests	0.81	0.79	0.80
Neural Network (100 hidden nodes, fully-connected)	0.71	0.72	0.72
Logistic Regression	0.61	0.60	0.62

6.6 Summary

In this chapter, supervised ML techniques were utilised to predict care outcomes for elderly patients who undergo hip fracture care in Ireland. Based on IHFD patient records, a regression model and classifier were developed to predict the inpatient LOS, and discharge destination respectively. A set of learning algorithms were experimented in this regard, and Random Forests proved to provide the highest prediction accuracy for regression and classification tasks. The developed predictors are claimed to make predictions on those outcomes with high accuracy. Further, the prediction models were deployed as predictive web services provided by the Azure platform.

Chapter 7

Machine Learning-Guided Simulations Applied to Healthcare Scenarios

7.1 Introduction

The practice of building simulation models has been largely dominated by knowledge articulated by domain experts. That knowledge can decide to a great extent the behaviour of a simulation model in terms of structure, assumptions and parameters. However, systems that deal with Big Data scenarios place further burdens on the modelling process, which can go beyond the capabilities of humans in many situations. For instance, the knowledge characterising a system may need to be derived from huge amounts of data, which may be accumulating with a high velocity as well.

In this Chapter, it is aimed to demonstrate how simulation models can be integrated with ML predictions. The main goal here was to provide a practical scenario where simulation models can be guided in concert with knowledge learned with the aid of ML. In conjunction with simulation experiments, ML models can be utilised to provide predictions that can guide the simulation model to mimic the actual system's behaviour. To demonstrate the practicality of the approach, the study employs a practical use case in relation to healthcare based on the IHFD dataset. Realistic scenarios were developed in the context of discharge planning for elderly patients with application to hip fracture care in Ireland.

A hybrid approach was embraced that integrated Simulation Modelling with ML. On one hand, an SD model was used to model the population level of patients. On the other hand, another DES model was used to model the elderly patient's journey through the care scheme of hip fracture. In tandem with the simulation model, predictive models were used to guide the simulation model. Specifically, the predictive models were used to make predictions on the inpatient LOS and discharge destination of simulation-generated patients. On a population basis, the simulation model provided demand predictions for healthcare resources related to discharge destinations, with a focus on long-stay care facilities such as nursing homes.

The simulation results suggested that there may be a need to reconsider the geographic distribution of nursing homes within particular areas in Ireland in order to keep abreast of the foreseen shift in demographics. Furthermore, the incorporation of ML within Simulation Modelling is claimed to improve the confidence in the simulation output.

7.2 Questions of Interest

A set of related questions were aimed by the study in relation to the care scheme of hip fractures. Particularly, the questions can classified into two categories as follows:

- Patient-level questions.
- Population-level questions.

Table 7.1 poses the questions in detail. In fact, the patient-level questions have been already addressed in Chapter 6, which included the development of supervised ML models to predict care outcomes (i.e. LOS, and discharge destination). The simulation model in this chapter would be integrated with these ML models as explained in the following sections.

Table 7.1: Questions of interest.

Scope	Questions
Patient-Level	Q1) How to predict the inpatient length of stay in acute facilities, given patient characteristics and care–related factors?
	Q2) How to predict the discharge destination, given patient characteristics and care–related factors?
Population-Level	Q3) What is the expected proportion of elderly patients discharged to home, or long-stay care facilities (e.g. nursing homes)?
	Q4) How adequate is the geographic distribution of long-stay care facilities with respect to the demographic profile of elderly population in Ireland?
	Q5) On a population basis, what is the expected utilisation of hospital resources (e.g. LOS) implied by hip fracture cases?

7.3 Approach Overview

The study embraced a multi-methodology approach that aimed to integrate simulation models with ML. In particular, the approach included four stages as follows: i) Knowledge discovery using unsupervised ML, ii) Modelling the dynamic flow of patients, iii) Modelling the care journey, and iv) Predicting care outcomes using supervised ML.

Reflecting on the questions of interest, the various stages were aimed to model the healthcare system at two inter-dependent levels (i.e. population level, and patient level). Initially at the population level, we utilised unsupervised ML using clustering techniques in an endeavour to discover potential underlying structures. Based on patient records, clustering and rule mining experiments were carried out to discover coherent patient clusters or frequent patterns. The knowledge discovery process was delineated in Chapter 4.

To provide a projected perspective of elderly patients discharged, SD modelling was used for that. The flow of elderly patients was modelled based on the discovered clusters. As such, each flow of elderly population represented a patient cluster of specific characteristics, and care-related outcomes.

Subsequently at the patient level, DES modelling was utilised in order to mimic the patient care journey at a finer-grained level of details. The DES model helped to deal with patient entities, rather than aggregate populations. Each patient entity could be treated individually in terms of characteristics (e.g. age, gender, type of fracture etc.), and care-related factors (e.g. time-to-admission, time-to-surgery etc.). Eventually, the ML models developed in Chapter 6 were utilised to make predictions on care-related outcomes for the simulated patients. For every simulated patient, ML models were used to predict the LOS and the discharge destination. Figure 7.1 illustrates the four stages of our approach.

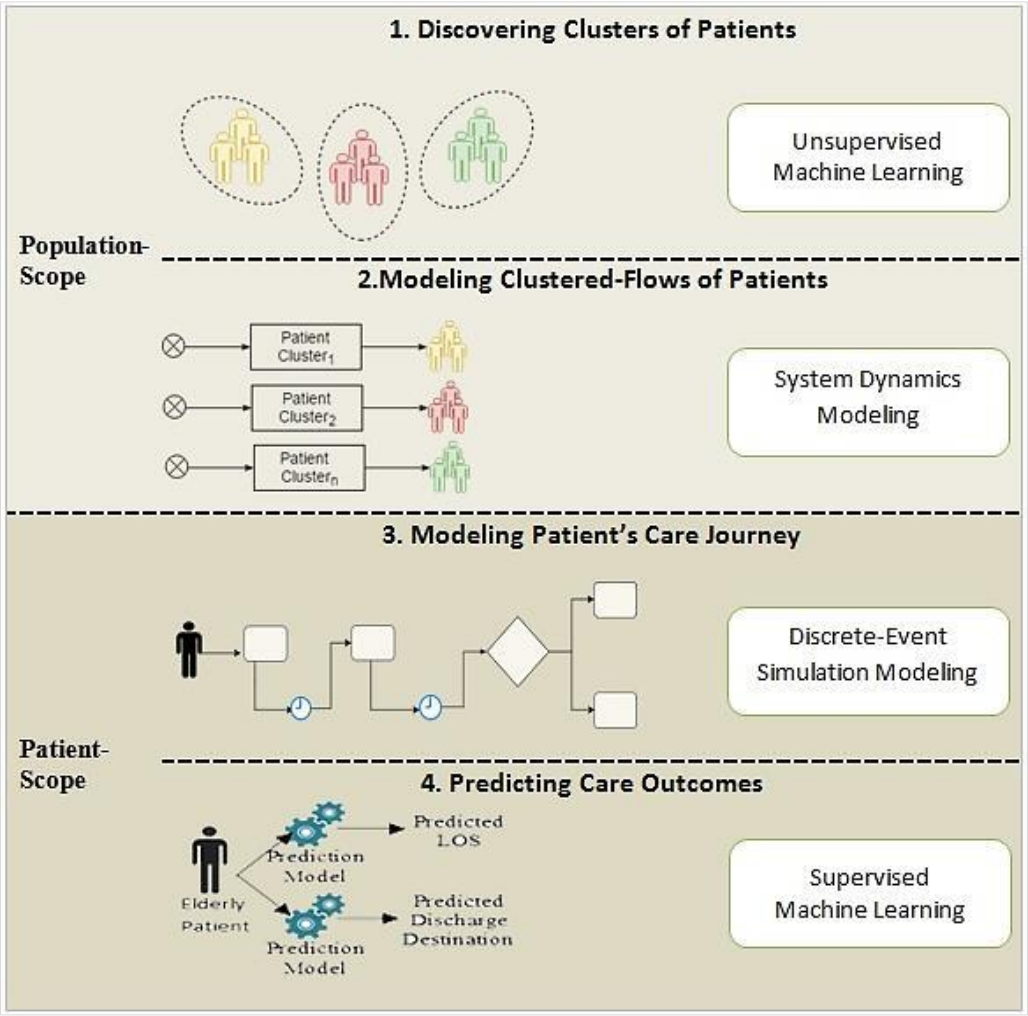


Figure 7.1: Approach overview.

7.4 The Care Journey of Elderly Patients

Simulation Modelling is both art and science with conceptual modelling lying more at the artistic end (Shannon, 1975). In this sense, the conceptualisation of the patient's journey represented an important step towards implementing the simulation model.

The patient's journey for hip-fracture treatment is composed of three main stages as follows: i) Admission, ii) Assessment and Treatment, and iii) Outcome. The journey typically starts from an admission source (e.g. home). Usually, the patient is initially admitted at the emergency department. Subsequently, the patient should be received at the orthopaedic ward. According to (British Orthopaedic Association, 2007), it was stressed that the primary surgery should be performed within 48 hours from admission, which can be an indicator of the care quality as well. Elderly patients usually undergo a number of assessments after surgery, which largely depends on the fragility history of each patient. Eventually, the patient is discharged to

home, or maybe a long-stay care facility (e.g. nursing home, or rehabilitation institute). Figure 7.2 sketches the patient journey from admission to discharge.

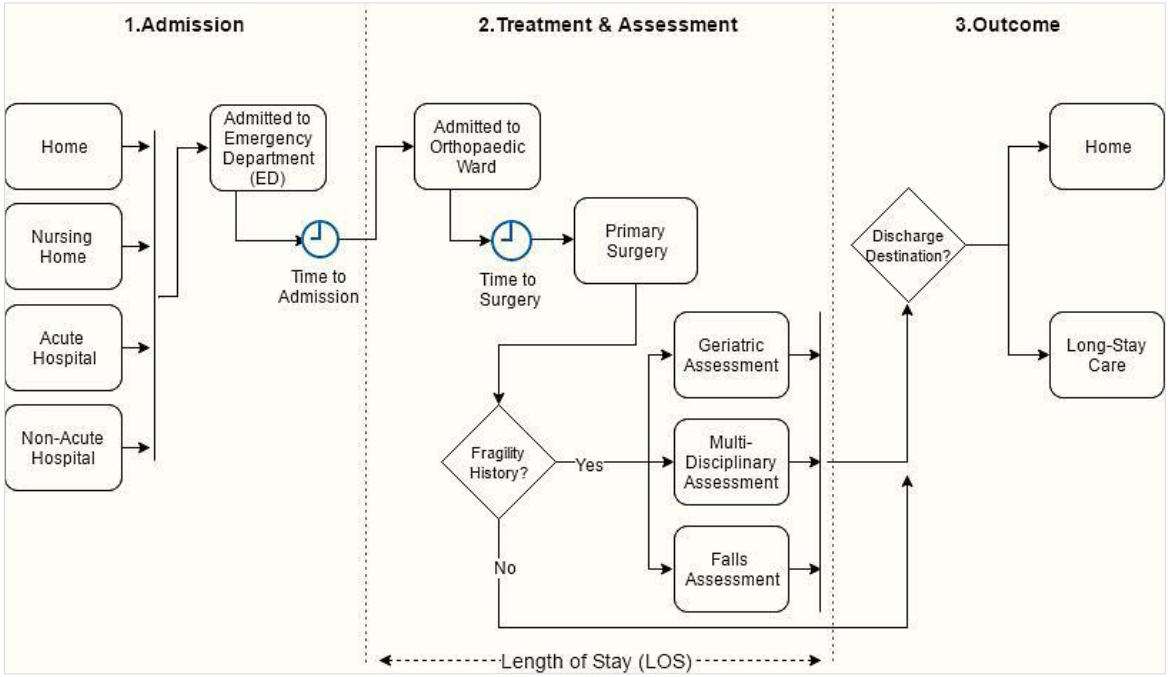


Figure 7.2: The care journey of elderly hip-fracture patients.

7.5 Assumptions and Simplifications

Effective conceptual modelling requires that the abstraction of a real or proposed system is an appropriate simplification (Pidd, 2003). In this regard, a set of assumptions and simplifications were decided while keeping the simulation model a sufficiently close approximation to reality for the intended application. Table 7.2 presents what assumptions or simplifications were made and why.

Table 7.2: Model assumptions and simplifications.

Assumption/Simplification	Purpose/Reason
The rate of hip fractures in population aged 60 and over was set as 407 for females and 140 for males per 100K.	The rate was defined by (Dodds, Codd, Looney, & Mulhall, 2009).
Elderly patients were assumed as aged 60 and over, though usually considered as aged 65 and older (Rosenberg & Everitt, 2001).	To conform to the defined hip fracture rate, considering patients aged 60 and over.
The scenario of patient transfer during the treatment course was not considered.	Only for simplification, assuming that the treatment course was operated within a single acute hospital.
The same age distribution for male and female patients is used.	For the purpose of simplification, since both distributions were slightly different.
The model assumed a discrete uniform distribution for patient's fragility history, whether positive or negative fragility.	The original distribution was not appropriate, since three were missing values for more than 34% of the records.
The elderly population per CHO was computed by applying a (fixed) percentage of the nation-wide projected population on a yearly basis. For example, the elderly population of CHO1 was computed as 9.5% of the total elderly projected population in 2016, whereas 9.5% was the actual percentage in 2014.	Due to lack of population information with respect to the 9 CHOs. The study obtained the population profiles of the CHOs for the year 2014 only.

7.6 Population-Level Modelling: System Dynamics

Initial System Dynamics Model

The initial model was intended to provide a bird's-eye view of the care scheme of hip fracture. The model mainly aimed at capturing the relationships among system entities in an SD fashion. However, this preliminary version did not consider the different characteristics of patients learned by the ML clustering experiments.

Specifically, the model focused on describing the dynamic behaviour pertaining to the continuous growth of population ageing, and the consequent implications on the incidence of hip fractures among the elderly. The model defined the main system actors as follows: i) Elderly patients, ii) Acute hospitals, and iii) Discharge destinations (e.g. home or long-stay care facilities). The model included a single reinforcing loop implied by patients with a fragility history, who are susceptible to re-sustain hip fractures or fall-related injuries. In this regard, it was reported that one in three of elderly patients fall every year, and two-thirds of them could fall again within six months (HSE, 2008). Figure 7.3 illustrates the initial SD model. Table 7.3 and Table 7.4 give the model variables, and equations respectively.

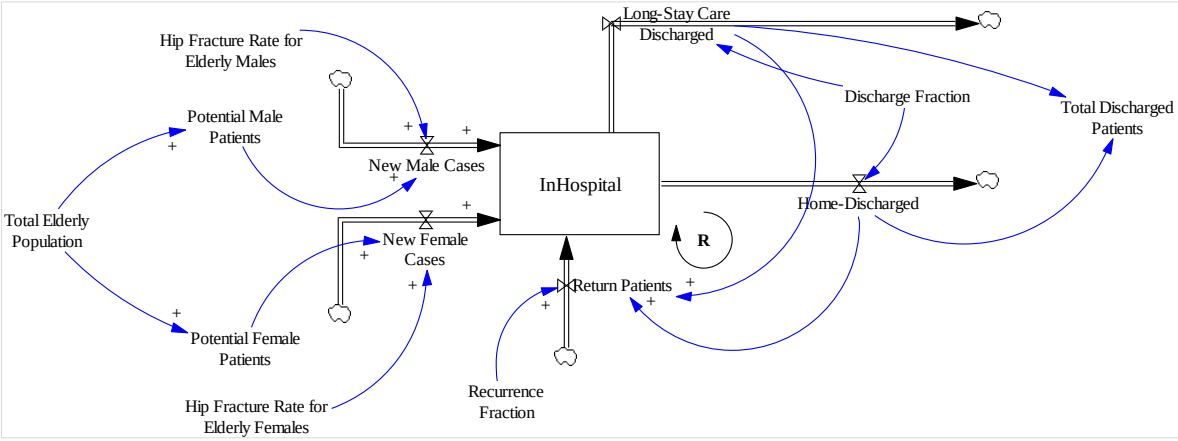


Figure 7.3: Initial SD model.

Table 7.3: Model variables.

Variable	Description
Total Elderly Population	The number of elderly population, aged 60 and over, in a particular year.
Potential Male Patients	Total male patients aged 60 and over.
Potential Female Patients	Total female patients aged 60 and over.
Hip Fracture Rate for Elderly Males	The rate of hip fractures in the total elderly male population = 140 cases per 100K.
Hip Fracture Rate for Elderly Females	The rate of hip fracture in the total population aged 60 and over = 407 for females per 100K.
InHospital	Stock variable representing the total number of elderly hip-fracture patients in acute hospitals nationwide.
Discharge Fraction	Proportion of total elderly patients discharged to home or long-stay care.
Total Discharged Patients	Represents the number of patients discharged to home and long-stay care.
Recurrence Rate	The rate that defines the proportion of discharged patients who are susceptible to re-sustain a hip fracture.

Table 7.4: Model equations.

Equation	Type
(1) Hip Fracture Rate for Elderly Males = 140 cases per 100K	Auxiliary
(2) Hip Fracture Rate for Elderly Females = 407 cases per 100K	Auxiliary
(3) New Male cases = Hip Fracture Rate for Elderly Males * Potential Male Patients	Inflow
(4) New Female Cases = Hip Fracture Rate for Elderly Females * Potential Female Patients	Inflow
(5) Home-Discharged= InHospital * Discharge Fraction	Outflow
(6) Long-Stay Care Discharged= InHospital * (1-Discharge Fraction)	Outflow
(7) Recurrent Patients = (Home-Discharged * Recurrence Rate) + (Long-Stay Care Discharged * Recurrence Rate)	Inflow
(8) InHospital =Integ((New Male cases+ New Female Cases) - (Home-Discharged + Long-Stay Care Discharged) + Recurrent Patients, Initial Value)	Stock

The Disaggregated Model

The SD model was re-designed in light of the clustering experiments conducted in Chapter 4. In particular, the model was disaggregated into three stocks representing the computed clusters of patients. Furthermore, the auxiliary variables were decided based on the cluster analysis. For instance, the first and second patient clusters were set to undergo the same TTS delay (i.e. TimeToSurgery1), while the third cluster was assigned a different delay (i.e. TimeToSurgery2).

Equally important, the inflows of elderly patients were structured based on the age variation within clusters. In particular, both of the first and third patient clusters were modelled to include more elderly patients (i.e. aged 80-100), while the second cluster was associated with less elderly patients (i.e. aged 60-80). This reflected the age groups within the patient clusters. Figure 7.4 illustrates the cluster-based SD model.

In this manner, the clustering model was employed effectively for the purpose of understanding the system structure, where the SD model stocks actually represented the three discovered clusters of patients. Moreover, the variations within clusters in terms of patient characteristics (e.g. age), or care-related factors (e.g. TTS) assisted with shaping the model behaviour. As such, it can be argued that the SD model was constructed with an established confidence predicated on the clustering model. The well-validated quality of clusters along with the compelling visualisations (see Chapter 4) could support the rationale behind the SD model design in terms of structure and behaviour as well.

The R language was used for implementing the SD model. The R-package *deSolve* (Soetaert, Petzoldt, & Setzer, 2010) facilitated solving the ordinary differential equations within the SD model. The R-script can be found in Appendix III (Code Snippets).

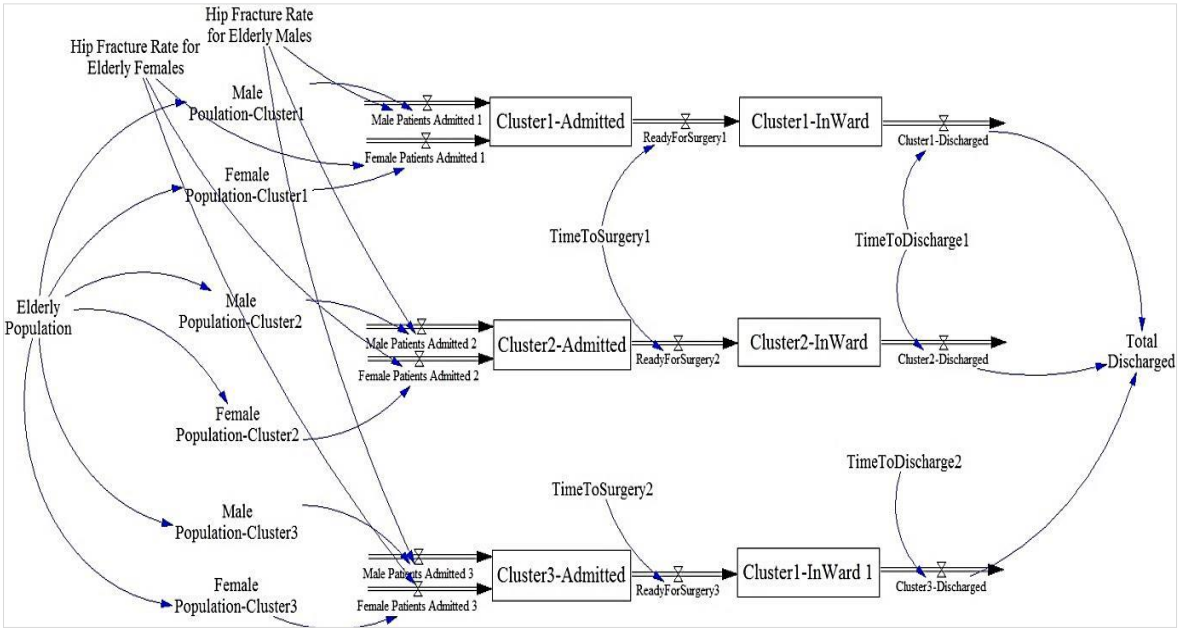


Figure 7.4: Cluster-based SD model.

7.7 Patient-Level Modelling: Discrete-Event Simulation

Simulation Approach

The DES approach was used to model a fine-grained perspective of the patient's journey. The model was utilised in order to produce a realistic sequence of events corresponding to those within the care journey, as sketched in Figure 7.2 earlier. Further, the DES approach can facilitate the following:

- Representing the system's components in terms of entities (e.g. patients, hospitals etc.)
- The entity-based modelling allowed for considering individual characteristics at the micro-level of the system. For instance, specific attributes of elderly patients (e.g. age, fracture type) were set differently based on the patient cohort (i.e. cluster).
- With such entity-based structuring, ML models can be accordingly used to provide predictions of care outcomes at the patient level.

Generation of Patients

The DES model made use of the projections produced by the SD model in order to generate individual patient entities. The generation process was implemented using the R language. The total number of generated patients reached around 30K for a simulated period of 10 years (i.e. 2017-2026). Table 7.5 presents the counts of elderly patients generated for every cluster.

Table 7.5: Counts of patients generated by the DES model.

Patient Cluster	No. of Simulation-Generated Patients
Cluster1	≈ 13K
Cluster2	≈ 11K
Cluster3	≈ 5K

Model Implementation

The simulation model was implemented using the DESMO-J framework (Lechler, & Page, 1999), a discrete event simulation library developed in Java. The DES model was mainly developed based on the empirical data acquired by the study. For instance, the probability distributions of patient attributes were set to mimic reality, as in the IHFD dataset.

The main entity of the simulation model represented the elderly patient. Each patient was assigned a set of attributes that characterised age, sex, area of residence, fracture type, fragility history and diagnosis type. The patients' characteristics varied based on the cluster they belonged to. Further care-related factors (e.g. TTS) were considered on an individual basis as well. Furthermore, the model endeavoured to blueprint the geographic structure of the Irish healthcare system by accurately mimicking the CHOs. Specifically, elderly patients were generated based on the CHOs' population profiles. Further, every CHO was structured as a set of acute hospitals and nursing homes, which were specified with accurate bed capacity using information from the HIQA organisation (HIQA, 2016).

The experimental environment consisted of two integrated parts. The DES model served as the core component. In tandem with the simulation model, ML models were then utilised to carefully predict the inpatient LOS and discharge destination for each elderly patient generated by the simulation model. The predictions were obtained from the ML models via web services enabled by the Microsoft Azure platform.

Figure 7.5 illustrates the environment of simulation experiments where the DES model was integrated with predictions from the ML models. In this manner, the ML models were employed to guide simulation experiments regarding the LOS and discharge destination of simulation-generated patients. The source code can be fully accessed through the GitHub repository (Elbattah, 2017).

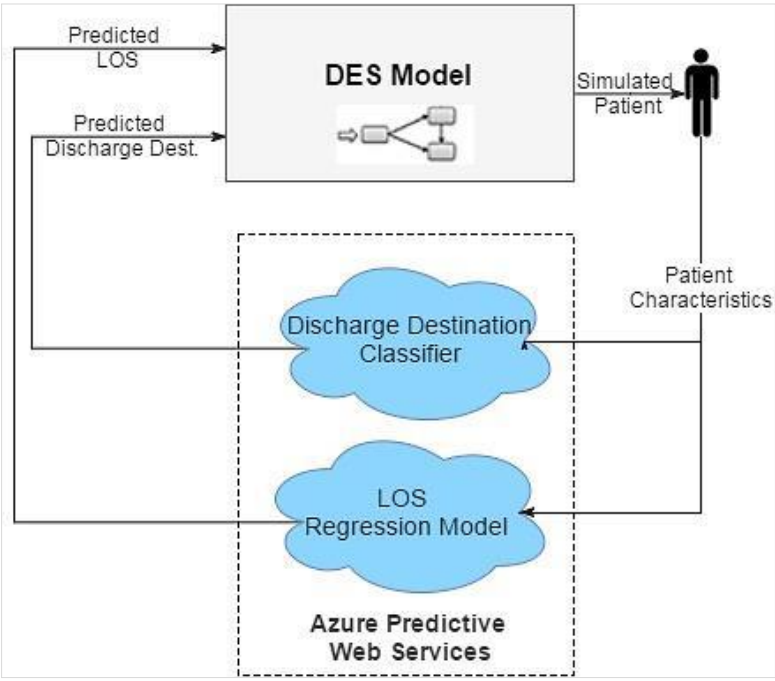


Figure 7.5: The experimental environment.

7.8 Results and Discussion

In line with the questions of interest, the simulation model output was interpreted in terms of the following: i) The expected number of discharged patients, ii) The inpatient LOS, and iii) The expected demand for long-stay care facilities as a discharge destination.

On one hand, Figure 7.6 jointly plots the projections of elderly male and female patients from 2017 to 2026. As expected, the discharged female patients constantly surpassed the male ones due to the pre-set disparate rates of arrival. On the other hand, Figure 7.7 shows the projected elderly patients with regard to the discharge destinations. The model expected that patients discharged to long-stay care can significantly be dominant over home-discharged patients. Furthermore, the demand for long-stay care was expected to keep growing over the simulated period. This expectation agreed with the official report (HSE, 2008) that acknowledged that less than one-third of hip-fracture patients go directly home after their hospital treatment.

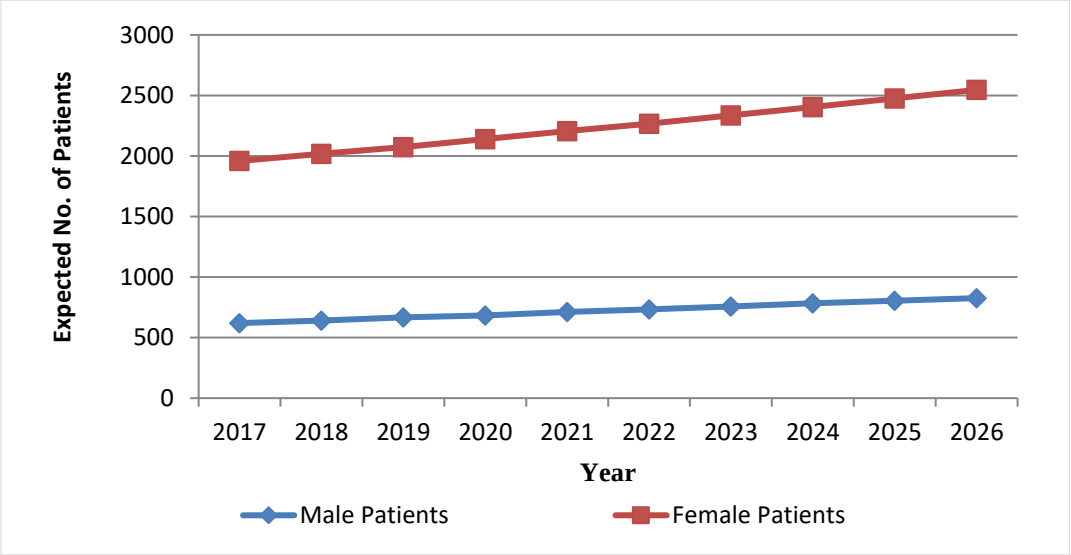


Figure 7.6: Projections of elderly patients expected to sustain hip fractures.

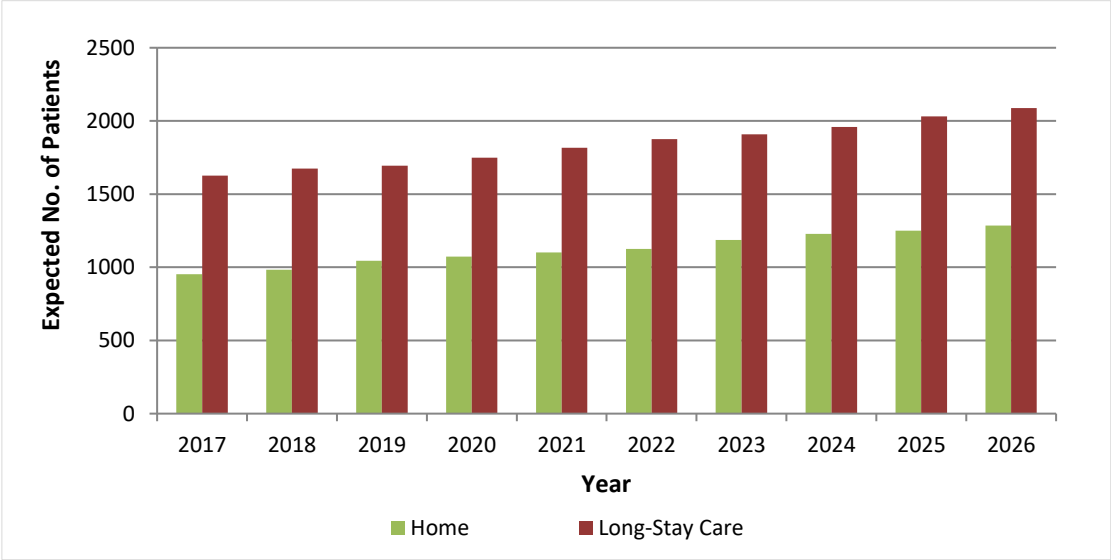


Figure 7.7: Projected elderly patients with respect to discharge destinations from 2017 to 2026.

Figure 7.8 aims to refine the model results with regard to every CHO individually. Given the 9 sub-plots in Figure 7.8, it could be concluded that most CHOs can likely have higher demands for long-stay care as a discharge destination. Specifically, CHO4 and CHO7 had the highest expected demand for long-stay care, which conformed with that CHO4 and CHO7 comprised the highest elderly population nation-wide, as explained in the background (see Chapter 2). On the contrary, CHO2, CHO5 and CHO8 were expected to experience higher levels of home-discharged patients, and fewer demands for long-stay care.

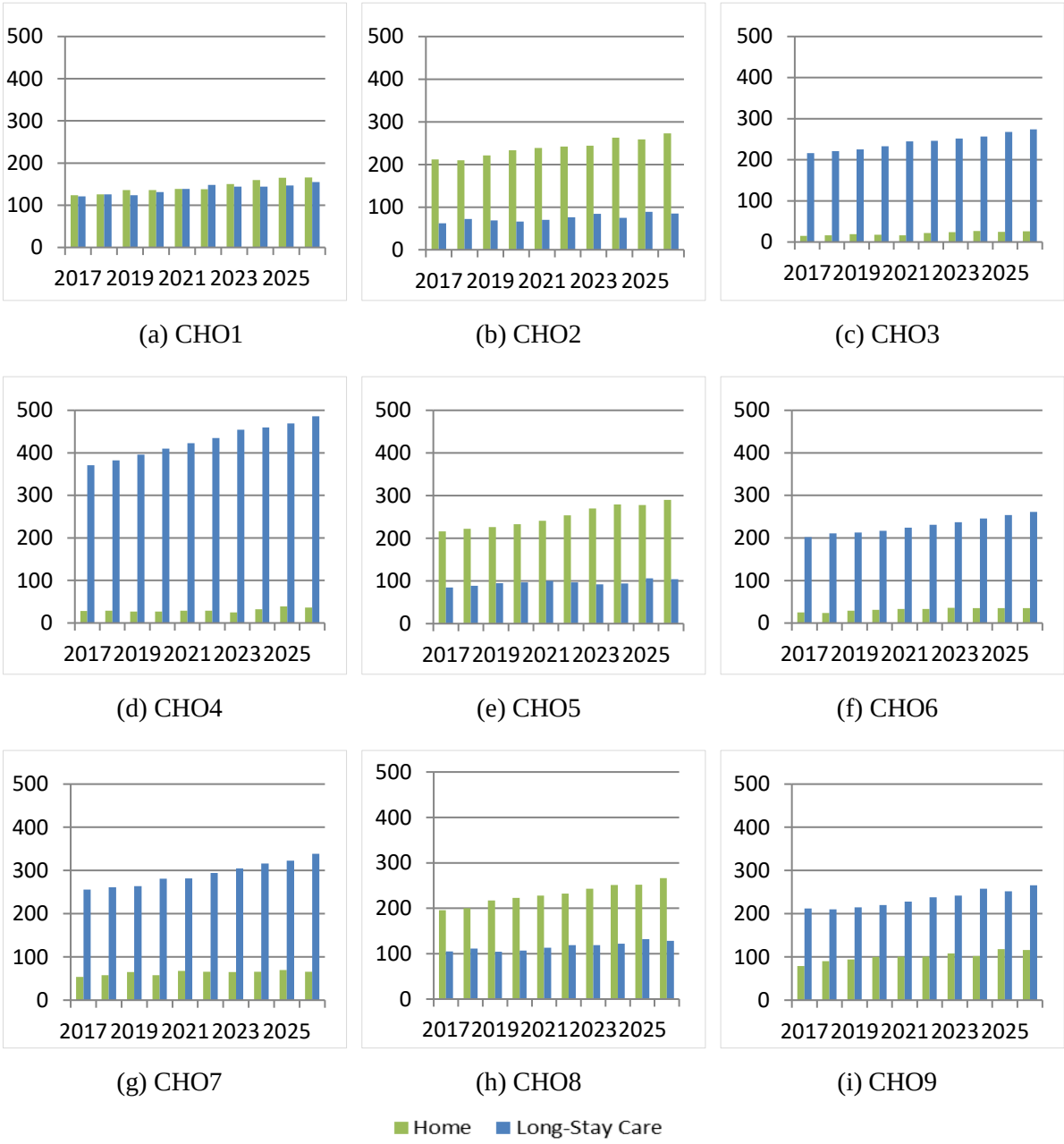
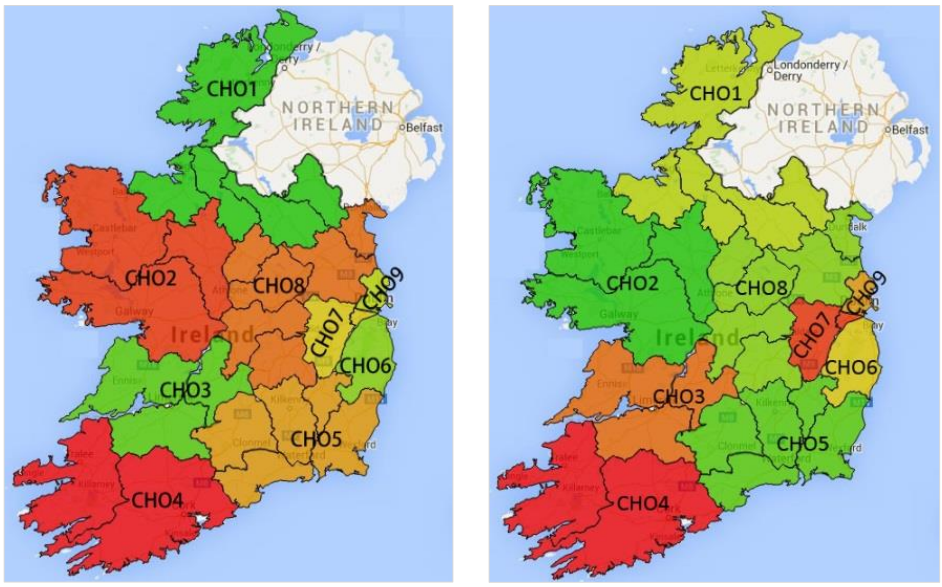


Figure 7.8: Predicted demand for discharge destinations with respect to the 9 CHOs individually from 2017 to 2026.

Further observations could be drawn in relation to the capacities of long-stay care facilities in the CHOs. Nursing homes were used as a concrete example of long-stay care resources due to the availability of information on their capacities nationwide. However, other facilities of long-stay care may exist such as rehabilitation institutions. Comparing bed capacities of nursing homes against the predicted demand, it turned out that a relative capacity-demand discrepancy might exist within some CHOs. For instance, CHO2, CHO5 and CHO8 were predicted to experience the lowest levels of demands, though they were reported to have significantly higher bed capacities. Furthermore, CHO4, CHO7 and CHO3 might probably be in need of higher bed capacities in the future, whereas they were predicted to have the top levels of demands for long-stay care. Figure 7.9 visualises a heat map of the 9 CHOs with regard to the current bed capacity and expected demand produced by the simulation model. Similarly, Figure 7.10 comparatively plots the bed capacity and the accumulative demand for long-stay care within every CHO.



(a) Bed capacity (as in 2017). (b) Predicted demand.

Figure 7.9: Heatmap representing bed capacity of nursing homes against predicted demand for long-stay care. Figure (a) visualises the current bed capacity of the 9 CHOs with reference to nursing homes as an example of long-stay care. The bed capacity of a given CHO is visually indicated by red (high) and green (low) in Figure (a). Figure (b) visualises the expected demand for long-stay care in every CHO. The predicted demand of a given CHO is visually indicated by red (high) and green (low). The relative mismatch in colours between Figure (a) and Figure (b) might infer an inconsistency of the geographic distribution of nursing home.

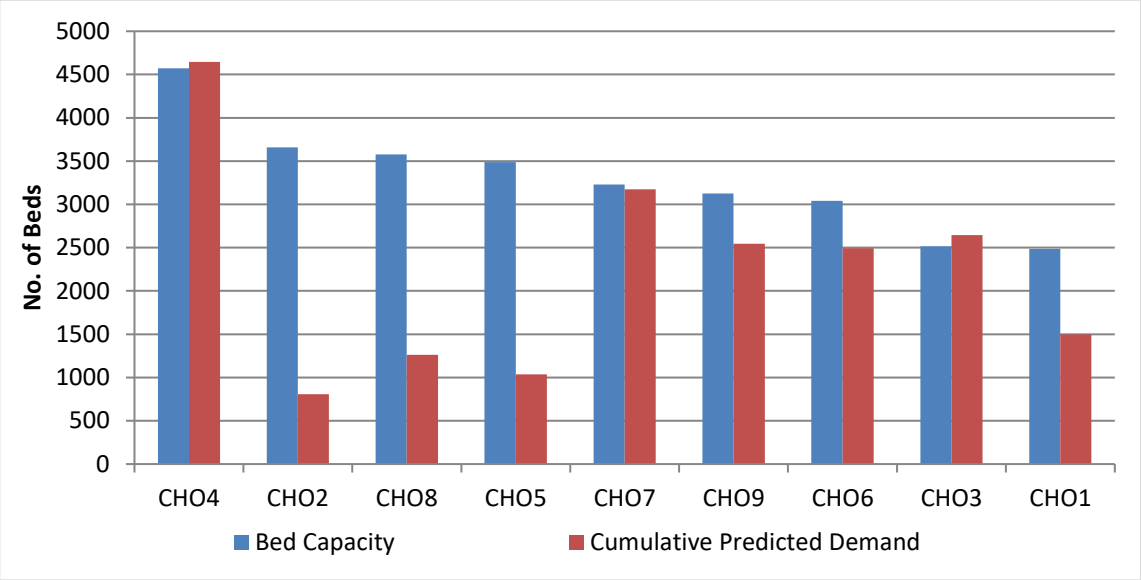


Figure 7.10: Capacity-demand analysis with respect to long-stay care in every CHO. The bed capacities of nursing homes are used as an example of long-stay care facilities.

The following figures interpret the simulation results with respect to the clusters of patients. Figure 7.11 plots the projections of elderly patients discharged from 2017 to 2027. It can be clearly observed that Cluster1 and Cluster2 steadily included the largest proportions of elderly patients. This adequately corresponded to the three clusters discovered by the ML clustering experiments, as shown earlier in Chapter 4.

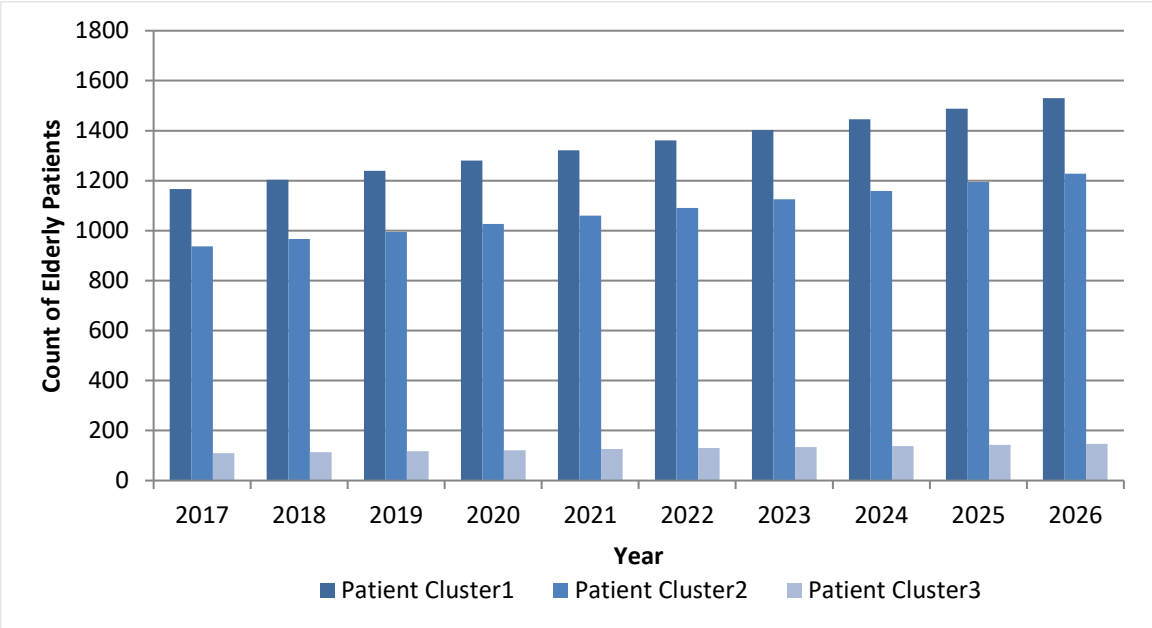
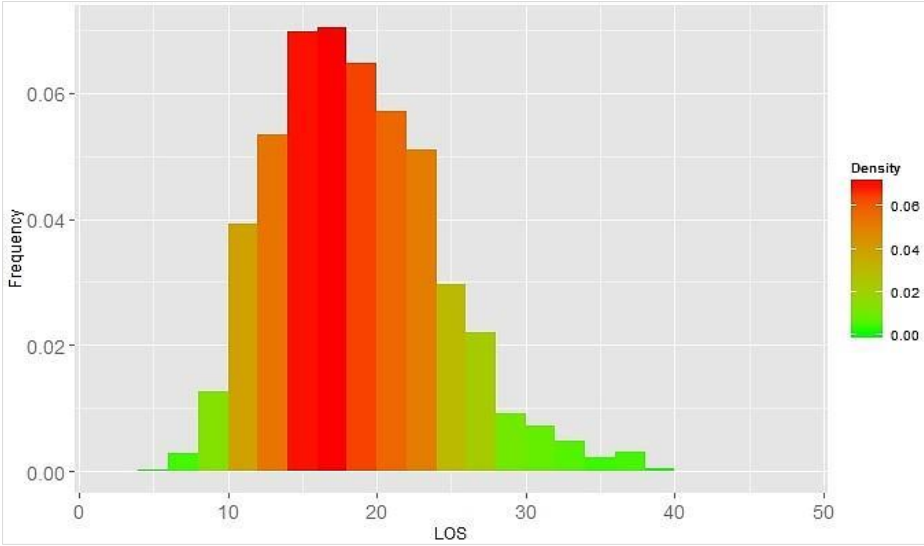
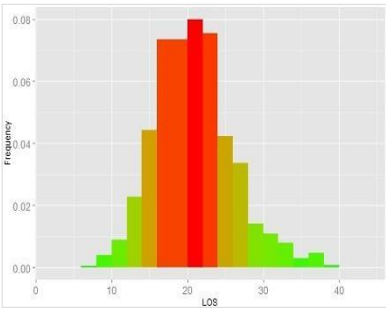


Figure 7.11: Cluster projections.

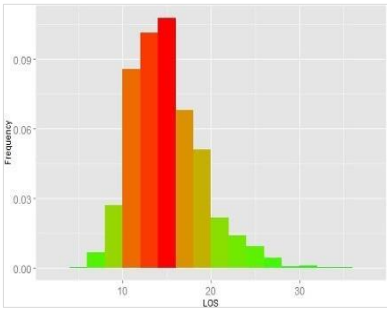
Figure 7.12(a) plots a histogram for the overall LOS expected over the simulated period for the three patient clusters. The histogram shows that the majority of patients experienced an LOS in the range of 10 to 30 days. In this regard, the model seemed to largely mimic reality, especially after excluding outliers. Moreover, Figure 7.12(b), Figure 7.12(c), and Figure 7.12(d) show the LOS with respect to every cluster individually. It turned out that Cluster1 and Cluster3 shared a similar distribution of the inpatient LOS, which to tended to be relatively longer compared to Cluster2 patients. The similar LOS distribution might be due to including the same age group (i.e. aged 80-100). This can translate into the importance of considering early intervention schemes for the category of more elderly patients.



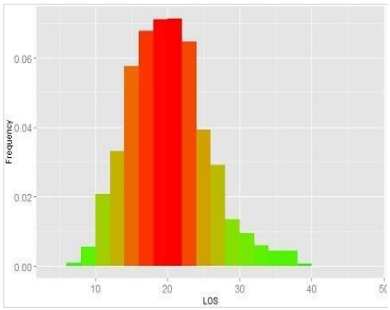
(a) Overall LOS.



(b) LOS-Cluster1.



(c) LOS-Cluster2.



(d) LOS-Cluster3.

Figure 7.12: LOS experienced for the simulated patients.

7.9 Model Verification and Validation

Model Verification

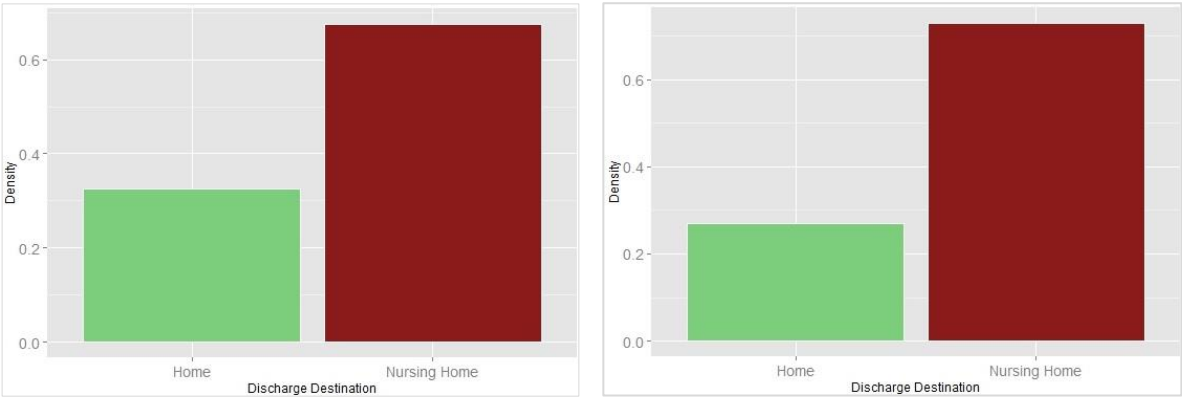
In order to validate the logic and suitability of the simulation model, a set of verification tests (Martis, 2006) were conducted throughout the simulation model's development, as follows:

- **Structure-Verification Test**: The model structure was checked against the actual system. Specifically, it was verified that the model structure was a reasonable representation of reality in terms of the underlying patient clusters, and associated elderly populations.
- **Extreme Conditions Test:** The equations of the simulation model were tested in extreme conditions. For example, the flows of elderly patients were set to exceptional cases (e.g., no elderly population aged 60 or over).
- **Parameter-Verification Test:** The model parameters and their numerical values were inspected if they largely corresponded to reality. Specifically, the probability distributions of patient attributes (e.g. age, sex and fracture types) were compared against those derived from the IHFD dataset.

Model Validation

According to (Law, 2008), the most definitive test of a simulation model's validity is comparing its outputs to the actual system. Similarly, we used the distribution of discharge destinations as a measure of the approximation between the simulation model and the actual healthcare system.

On one hand, Figure 7.13 provides a histogram-based comparison between the actual system and the simulation model regarding the discharge destination. The comparison showed that the distributions of the actual system and simulation output were largely similar. However, the comparison revealed that the model slightly underestimated and over-estimated the proportion of patients discharged to homes, and long-stay care facilities respectively.



(a) Actual system. (b) Simulation model.

Figure 7.13: Histograms of the discharge destination in the actual system and simulation model.

On the other hand, Figure 7.14 compares the actual system's average LOS to that of the simulation model with respect to the 9 CHOs separately. The figure clearly shows that the simulated CHOs' average LOS matched the actual system very well, without any significant over- or under-estimation. Overall, validation and verification tests proved that the simulation model can be suitable for answering questions from the perspective of the study objectives.

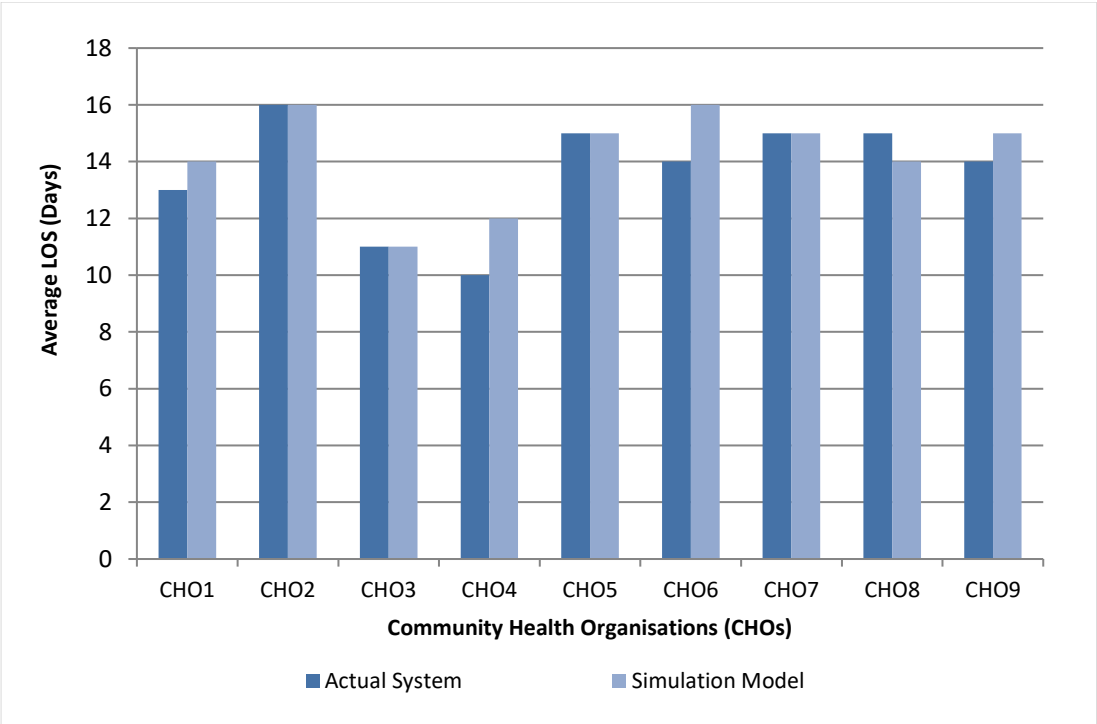


Figure 7.14: CHO-based comparison between the actual system and simulation model in terms of average LOS.

7.10 Summary

The chapter presented a hybrid approach that integrated Simulation Modelling with ML including unsupervised and supervised techniques at different stages of model development. A practical scenario was presented in relation to healthcare planning in Ireland to demonstrate the applicability of the approach. First, the knowledge learned by unsupervised ML models in Chapter 4 was used to assist the modelling phase. Second, simulation experiments were conducted with the guidance of ML models trained to make predictions on the system's behaviour. The core idea was to realise ML-guided simulations during the phases of model design or experimentation.

From a practical standpoint, the chapter also delivered useful insights in relation to the expected demand for hip fracture care due to population ageing. The model outputs realised a population-based perspective of the demand for hip fracture care, with a focus on elderly patients in particular. The projections of elderly discharged patients were compared to the present bed capacities of nursing homes, as an ideal example of long-stay care.

The insights were provided based on a well-rounded picture corresponding to the demographic profiles, structure, and capacity of the healthcare system in Ireland. The results revealed that the current geographic distribution of nursing homes may not match with the projected demographic profile of elderly patients within particular regions (i.e. CHOs) in the future. The incorporation of ML is claimed to yield further confidence in the simulation model, and in turn improving its credibility for decision making.

Chapter 8

Adaptive Simulation Models Aided By Machine Learning

8.1 Introduction

This chapter follows on the path of exploring the possibilities of integrating simulation models with ML. It was mainly aimed at realising the idea of 'self-adapting' simulation models. The premise was that the behaviour of a simulation model can be adaptive by the guidance of ML models, which are being incrementally trained to predict the actual system's behaviour. In this manner, simulation models can learn to change their behaviour in accordance with changes in the behaviour of the system of interest.

As a proof-of-concept, a set of scenarios were built based on the healthcare use case used in previous chapters. The scenarios applied a set of policy changes that were considered as changes in the system behaviour. In conjunction with simulation experiments, ML models are used to provide predictions guiding the simulation model to adapt its behaviour to new system's states.

8.2 Overview of Experiments

Practical experiments are conducted in this section for the purpose of demonstrating the applicability of the approach. The experiments were particularly built on the healthcare use case developed in previous chapters. Further hypothetical scenarios were developed to examine the simulation model behaviour to new situations in the real system.

Simulation Experiments

The experimenting environment consisted of two parts. On one hand, the DES model served as the core component used to generate patients and simulate their care journey. On the other hand, the ML models were then utilised to predict the inpatient LOS and discharge destination for each elderly patient generated by the simulation model. The predictions were obtained from the ML models via web services enabled by the Microsoft Azure platform. Figure 8.1 illustrates the environment of simulation experiments where the DES was integrated with predictions

from the ML models. Table 8.1 presents the counts of elderly patients generated over 50 simulation experiments for every year (2013-2015).

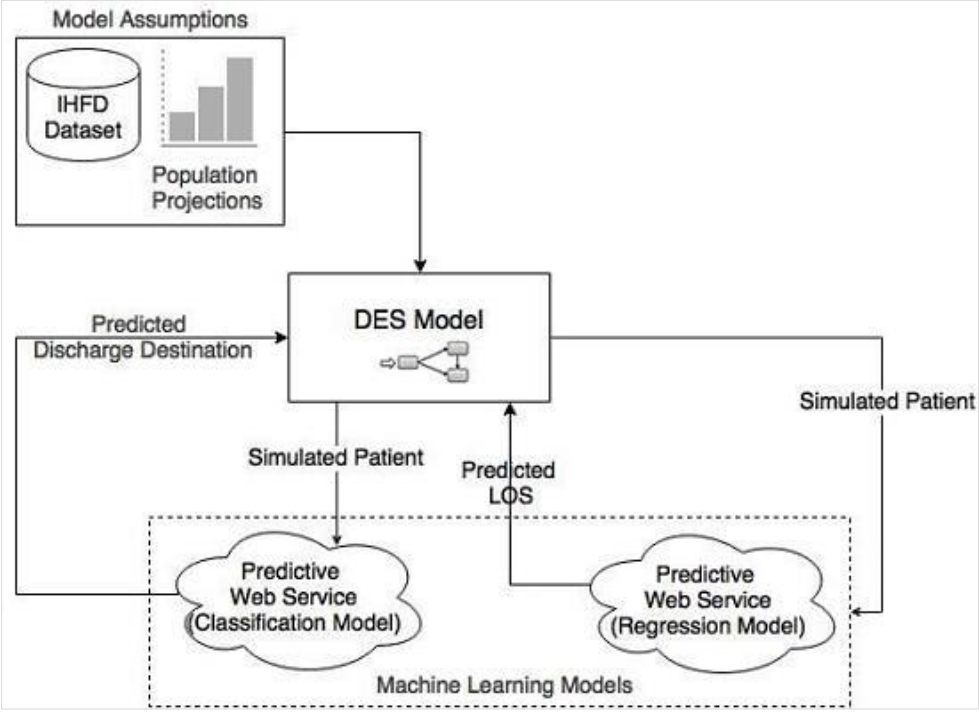


Figure 8.1: Simulation experiment aided by ML models.

Table 8.1: Counts of patients generated per year over 50 simulation experiments.

Year	No. of Generated Patients
2013	≈93K
2014	≈95K
2015	≈98K

Simulation Scenarios

The primary goal of the experiments was to validate the rationale that the simulation model could adapt to changes in the system's behaviour in an automated manner with the aid of ML models. In this respect, the experiments were designed to simulate two scenarios. The first scenario represented the base case, whereas the simulation model was only used to reproduce the flow of patients for the years 2013-2015. The base case was used to initially validate the simulation model output, which was also assisted by the ML predictions on the LOS and discharge destinations.

The second scenario was intended to simulate a hypothetical change in the system behaviour as follows. It was assumed that a new policy was introduced staring from the year 2014 in order to improve the patient's journey. The new policy aimed to maintain the hip-fracture care standards by keeping the TTA and TTS within 4 hours and 48 hours respectively. In accordance with the new policy, the average LOS of patients was assumed to decrease by 20% and 30% in the years 2014 and 2015 respectively. Further, the proportion of patients discharged to long-stay care was set to decrease by 5% and 10% for the years 2014 and 2015 respectively.

In order to reflect the new policy, the datasets of the years 2013 and 2014 were synthetically altered. For instance, the LOS was reduced by 20% for patients discharged in the year 2014. Further, the ML models were re-trained in view of the change in system behaviour. Eventually, the simulation experiments were re-run, and the output was compared against the base case behaviour. Table 8.2 compares LOS average values in the actual system before and after applying the new policy.

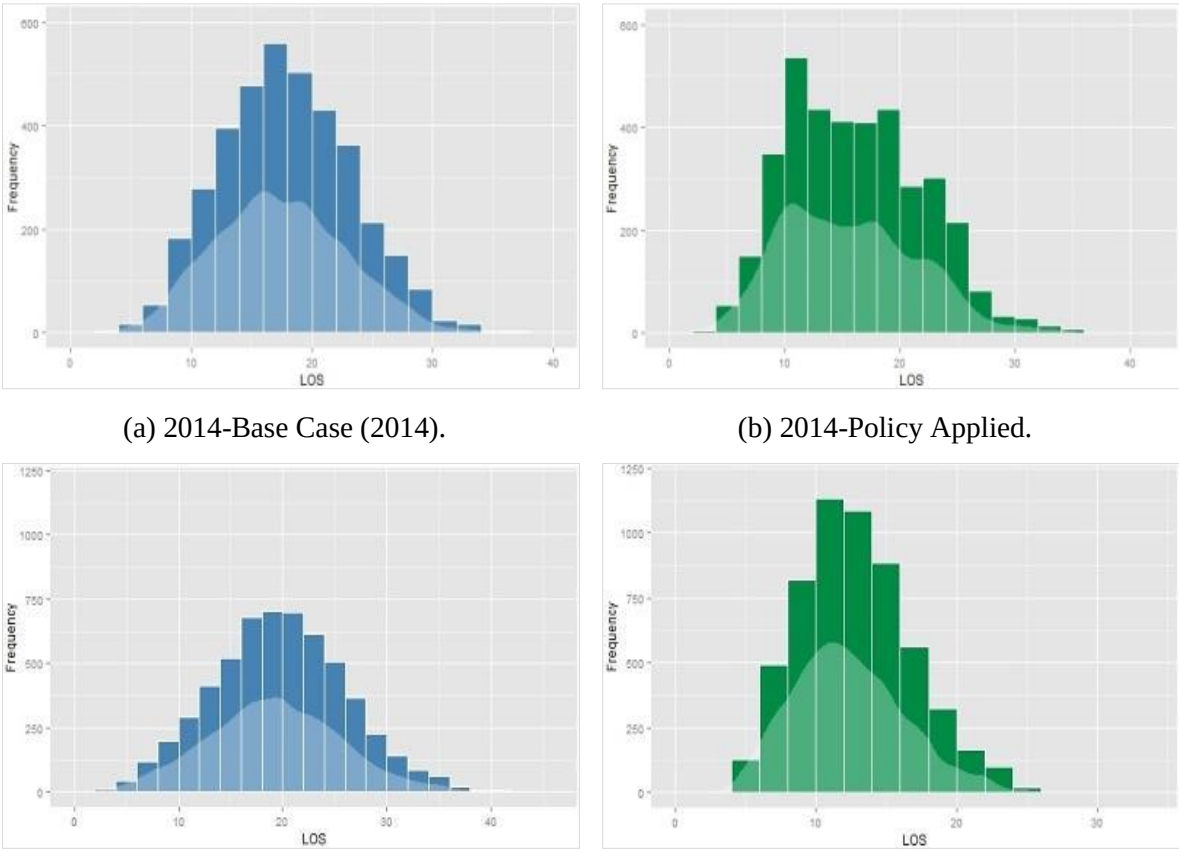
Table 8.2: Average LOS in the dataset before and after applying policy.

Average LOS (≃)	
Pre-Policy	**Post-Policy**
11.80494 Days	9.43155 Days

8.3 Results

This section investigates the model behaviour before and after applying the new policy for care improvement. Specifically, the behaviour of the simulation model is interpreted in terms of the following: i) The distribution of inpatient LOS, and ii) The expected proportions of patients discharged to home, or to long-stay residential care.

Figure 8.2 plots the LOS distribution for the years 2014 and 2015 before and after applying the new policy. In 2014, it can be observed that patients experienced lower LOS periods, which corresponded to the 20% improvement policy. Furthermore, the improvement was more pronounced in 2015, which corresponded to the policy that aimed to decrease the LOS by 30%. Further, Table 8.3 compares LOS average values in the simulation output before and after applying the new policy.



(a) 2014-Base Case (2014). (b) 2014-Policy Applied.

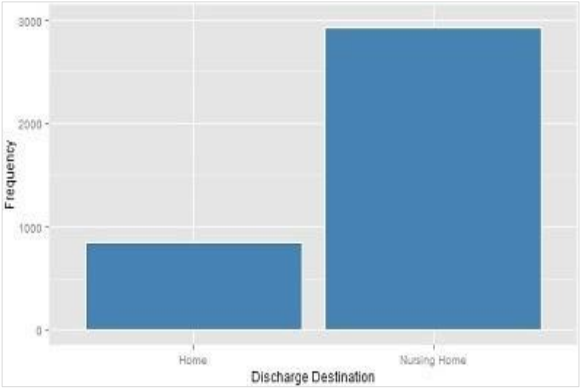
(c) 2015- Base Case. (d) 2015-Policy Applied.

Figure 8.2: Simulation output regarding the inpatient LOS. The blue plots represent the pre-policy simulation experiment (Base Case) in 2014 and 2015. The green plots represent the post-policy simulation experiments.

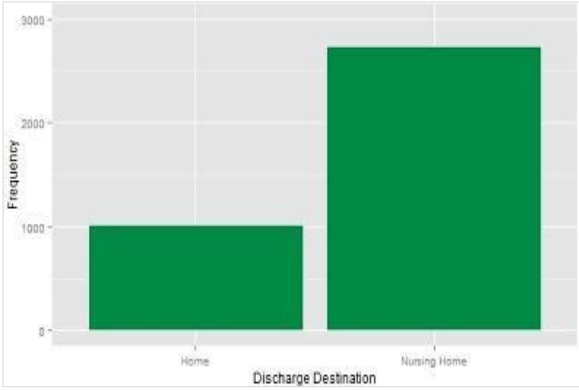
Table 8.3: Average LOS in the simulation output before and after applying policy.

Average LOS (≃)	
Pre-Policy	**Post-Policy**
19.234 Days	12.34278 Days

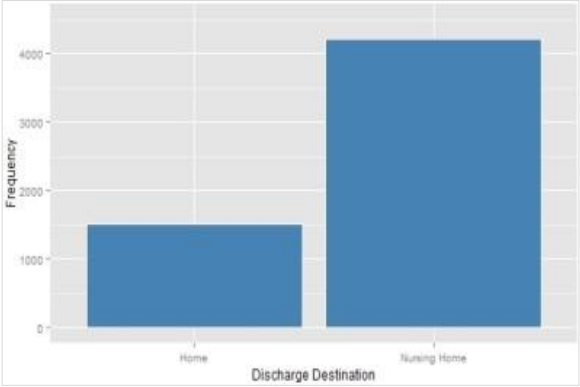
Likewise, Figure 8.3 shows the distribution of the discharge destinations. The figure shows that the year 2014 indicated an improvement of the proportion of home-discharged patients, which in turn reflected the new policy measures. The improvement is also more significant in 2015. The simulation results clearly showed that the model behaviour has adapted in accordance with the new knowledge informed by the ML models. Specifically, the new policy measures reflected the predictions of the ML models, which in turn modified the behaviour of the simulation model. This can validate the rationale that simulation models can adapt to changes in system behaviour in an automated manner with the aid of ML models.



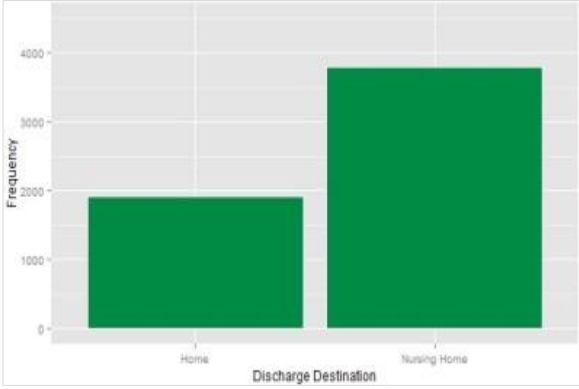
(a) 2014-Base Case (2014)



(b) 2014-Policy Applied



(c) 2015- Base Case



d) 2015-Policy Applied

Figure 8.3: Simulation output regarding the discharge destinations. The blue plots represent the pre-policy simulation experiment (Base Case) in 2014 and 2015. The green plots represent the post-policy simulation experiments.

8.4 Limitations

A set of limitations needs to be acknowledged as follows. First, the simulation scenarios were not based on real data, but a synthetically altered version of the IHFD dataset in order to simulate a hypothetical change in the system behaviour.

Another relevant issue of concern is that presented use case may not be the ideal scenario to demonstrate the potentials of integrating simulation modelling and ML. It is believed that a typical Big Data scenario could better present the benefits of that integration.

8.5 Summary

This chapter discussed the idea of realising adaptive simulation models aided by ML. Guided by ML predictions, it was demonstrated how a simulation model can adjust its behaviour to reflect on changes in the actual system behaviour. As such, simulation models can learn to change their behaviour in accordance with system's behavioural changes in an automated manner, or with minimal human input.

The chapter utilised the healthcare use case, simulation models, and ML models developed previously to validate the approach. The key idea was demonstrated based on hypothetical scenarios that served as a proxy for changes in the actual system conditions. In particular, the scenarios assumed changes in policy towards improving the care in terms of LOS. It was observed how the simulation model adapted its behaviour with the incremental training of ML models that provide predictions on the inpatient LOS and discharge destination.

Chapter 9

Conclusions and Future Directions

9.1 Conclusions

The integration of simulation models with ML can yield potential benefits for the practice of Modelling and Simulation. That integration can help address further complex questions and scenarios of analytics. In essence, if there is a need to achieve a complete representation of the real-world problem, then the integration with ML can provide more accuracy in the problem representation and solution. A hybrid modelling approach aided by ML can support the process of decision-making to gain experience by acting on realistic dynamic scenarios.

The study presented several examples of how ML can assist Modelling and Simulation at different stages of model development. To summarise, the data-driven knowledge gained by machine intelligence can play a key role in:

- Increasing the level of reality and fidelity of simulation models, which in turn brings further realism and intelligence to actors and scenarios in computer simulations.
- Extending the modelling capabilities at the micro-level interactions, associations, and inter-dependence among real-world entities. In the study case, for example, simulation modelling was used to represent patients at the population level, while ML-based predictions provided a focused perspective at the patient level in terms of care outcomes (e.g. LOS, and discharge destination).
- Lowering the bias of human-based elicitation of knowledge, which can in turn increase the confidence in simulation models. In this regard, the integration of data-driven insights learned by ML models can lower the subjectivity of the mental models of modellers, simulationists, or subject matter experts, which can in turn increase the confidence in simulation models.

- Realising adaptive simulation models that can learn to change their behaviour in accordance with system behavioural changes in an automated manner. This can be particularly beneficial for modelling in dynamic systems that inherently exist within rapidly changing environments, where the situation of concept drift can have an impact on the quality of model prediction over time.

In light of the experiments conducted, the study conceives a category of simulation models that can be aided by ML models, named as *ML-Aided Simulations*. ML-Aided Simulations can realise the learning factor based on incremental predictions provided by ML models. In this regard, the definitions of ML are rephrased to provide an interpretation of the study's viewpoint as below.

Table 9.1: ML-Aided Simulations.

Definitions of Machine Learning	**ML-Aided Simulations (Simulations + ML)**
The subfield of computer science that gives computers the ability to learn without being explicitly programmed (Samuel, 1959).	Simulation models given the ability to adapt to new system changes or knowledge without being explicitly informed by modellers.
Changes in the system that are adaptive in the sense that they enable the system to do the same task(s) more efficiently and more effectively the next time (Simon, 1983).	Changes in the simulation model that are adaptive in the sense that they enable the model to answer the question(s) of interest more efficiently and more effectively the next time.
Things learn when they change their behaviour in a way that makes them perform better in the future (Witten, Frank, & Hall, 2005).	Simulation models learn when they change their behaviour in a way that makes them mimic the system of interest better in the future

9.2 Answers to Research Questions

This sections examines how the study provided answers to the research questions as stated in Chapter 1. The questions along with answers are provided below.

Q1. How can ML be employed to assist the conceptualisation of systems?

The study presented arguments for considering ML as a valid path to assist the practice of modelling, especially in case of systems that are inherently encompassed with data of high complexity. With ML, the process of learning about systems can be conducted in a fully or partially automated manner. Further, ML can work effectively in situations where the system's behaviour can be largely learned by examples, rather than expressed analytically.

Through practical examples, it was demonstrated how ML can be utilised as an assistive artefact within the process of problem formulation. Unsupervised ML techniques can be effectively employed at the phase of conceptual modelling for the conceptualisation of the system's structure or behaviour.

Q2. Is it possible to integrate mental models with ML models in a way that supports the learning process to be developed based on a more data-driven manner? If so, how?

Yes, the study discussed a potential direction to achieve this goal. An approach was developed where mental models can be supported with data-driven knowledge learned by ML. The key idea was that mental models can be assisted by ML models trained to make predictions on a particular aspect of the system's structure, or behaviour under study. It is assumed that changes in the system sates or conditions can be inferred, at least partially, through ML predictions.

The approach considers new data generated by new states of the actual system as a form of feedback. Based on such data-driven feedback, ML models can be trained to predict the future behaviour of the system. Moreover, ML models can be continuously re-fitted to echo feedback loops, and reflect new system's conditions. In this manner, behavioural changes can be learned based on ML models in tandem with mental models. The approach was practically implemented in scenarios related to healthcare in Chapter 5.

Q3. Which ML techniques can be appropriate for the perception of the structure, or behaviour of systems involved within a problem?

The study indicated that unsupervised ML can adequately serve the purpose of knowledge elicitation with minimal prior assumptions. Unsupervised techniques (e.g. clustering, rule mining) can be employed at the phase of conceptual modelling as an assistive artefact for the conceptualisation of the system's structure or behaviour. For instance, data-driven insights gained by a clustering model may help identify significant structures or patterns underlying the system of interest. This can in turn reflect on the structure and design of the simulation model. Similarly, unsupervised ML can be used to explore and analyse data output from simulation experiments.

Q4. Can the integration of ML lead to a higher level of confidence in simulation models, indicated by the accuracy of ML models?

The integration with ML may help simulations attain a higher level of model realism. The study presented its view on how ML can be employed as an assistive tool to help reduce the epistemic uncertainty (Oberkampf et al., 2002) underlying simulation models. That kind of uncertainty is largely attributed to the subjective interpretation of system knowledge based on human inputs. In other words, if simulation models could be supported by predictive models trained to make predictions on the actual system's behaviour, this could in turn lead to a relatively lower degree of uncertainty. Accurate ML predictions can arguably extend the fidelity of modeling with regard to the micro-level behaviour of simulation entities or agents

Q5. How simulation models can learn about changes in the system's behaviour or conditions with minimal involvement of modellers or simulationists?

Supervised ML techniques can be utilised at the experimentation phase. ML models can be embedded into simulation experiments to realise the learning factor. Concurrently with a simulation experiment, the simulation model can be guided by ML models trained to predict the system's behaviour. Such predictions would suggest the behaviour at the micro-level of the system under study, as entities or agents in DES or ABS models. Chapter 3 developed a conceptual framework to realise the idea of ML-guided simulation experiments.

Chapter 8 also discussed the idea of realising adaptive simulation models aided by ML. Guided by ML predictions, it was demonstrated how a simulation model can adjust its behaviour to reflect on changes in the actual system behaviour. As such, simulation models can learn to change their behaviour in response to system's behavioural changes in an automated manner, or with minimal human input.

The approach rationale was validated based on hypothetical scenarios that served as a proxy for changes in the actual system conditions. It was observed how the simulation model adjusted its behaviour in accordance with the incremental training of ML models providing predictions on particular variables, which have a considerable influence regarding the questions of interest.

9.3 Future Directions

Looking into the future, more sophisticated ML techniques can be utilised in order to distil the knowledge underlying further complex systems or scenarios. For example, it would be interesting to investigate how simulation models can be integrated with Deep Learning. The development of Deep Learning (LeCun, Bengio, & Hinton, 2015) received wide attention within the ML research for its capacity that dramatically improved the state-of-the-art in hard problems. Deep Learning models are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. The multiple processing layers can effectively represent linear and non-linear transformations.

The study conceives that simulation models can be assisted by predictions from Deep Neural Networks (DNN) trained to capture the system knowledge in a mostly automated manner. For instance, Figure 9.1 illustrates a simulation model where a variable (i.e. Predicted Variable) is input to the model. That variable can be predicted using a DNN, which was trained to capture the new system state. In this way, the DNN can be continuously trained in case of the arrival of new data that echo new system states or conditions. Similarly, Figure 9.2 views a set of DNNs used to guide the behaviour of a simulation model. In a broader sense, the need for constructing a high-fidelity representation of real-world problems shall bring up new practical aspects for taking advantage of ML potentials.

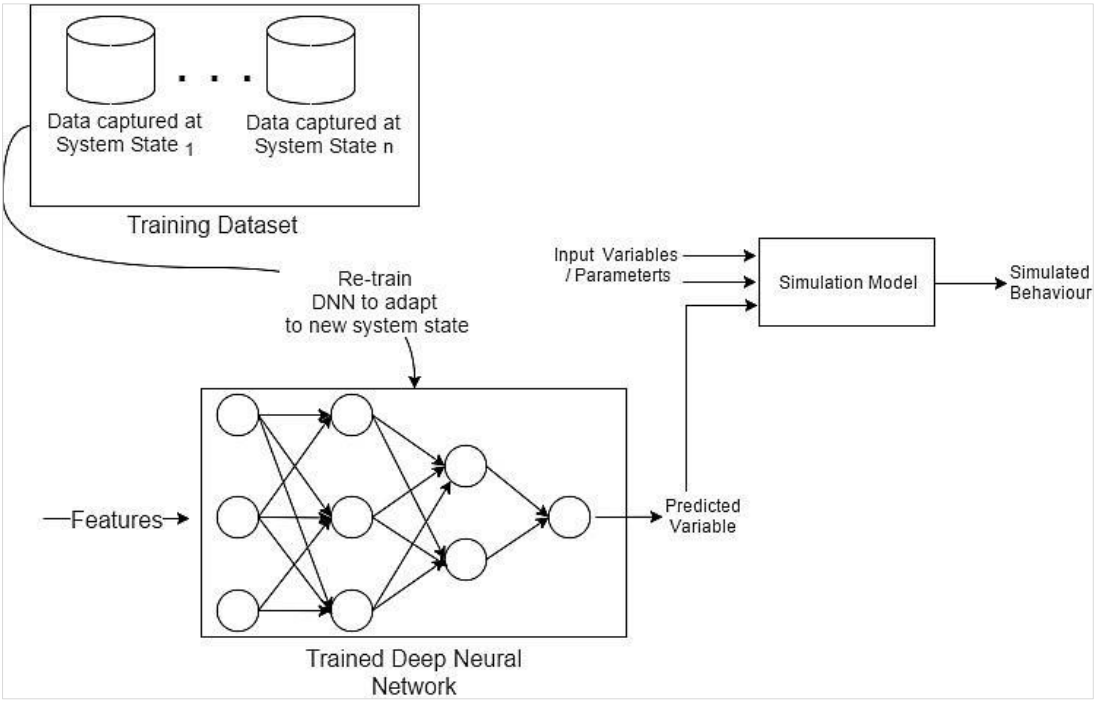


Figure 9.1: Adapting to the new system states via predictions from a trained DNN.

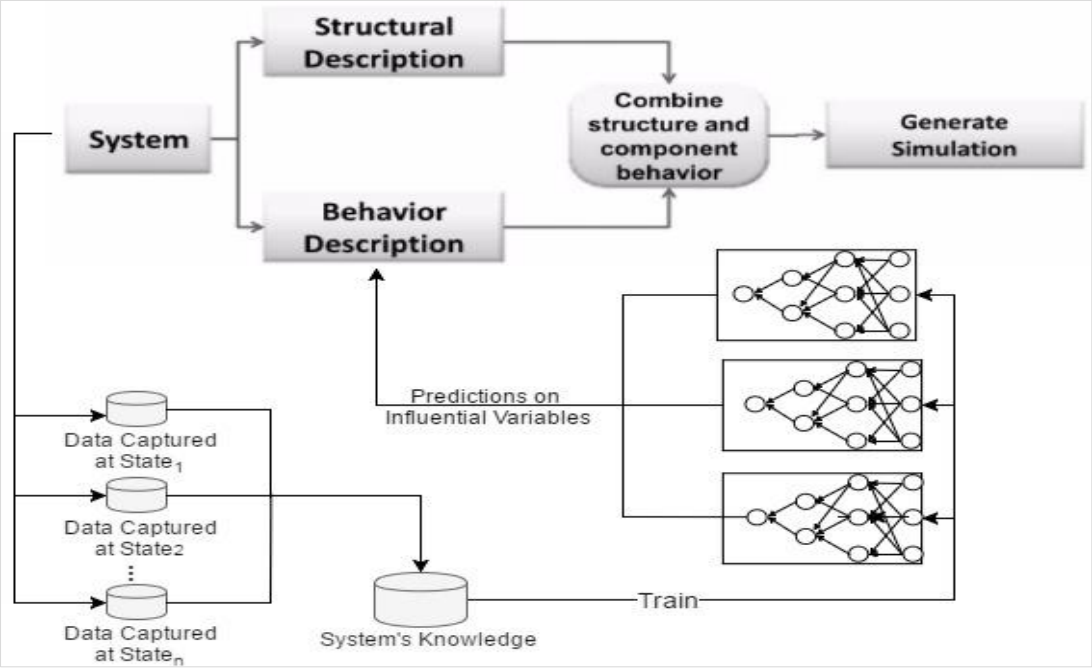


Figure 9.2: Hybrid systems modeling aided by DNN.

Appendix I: IHFD Dataset Description

Table: Dataset Variables

Variable	Variable Name in Database	Data Type	Description
Area of Residence	RESID	Integer	The place the patient would normally reside. Foreign nationals resident in Ireland should have a code assigned for their Irish place of residence. Foreign visitors on short stay should be coded to their country of residence.
Date of Birth	DOB	Date	Patient's Date of Birth. At national level only month and year of birth are available with day of birth set to 15.
Date of Discharge	DISDATE	Date	The date on which the patient is discharged from hospital following an episode of care. If the patient dies during the episode of care, the date of death is the date of discharge.
Date of Principal Procedure	PROCDATE1	Date	Date the Principal Procedure was performed.
Date of procedure sequenced 2nd	PROCDATE2	Date	Date of procedure sequenced 2nd.
Date of Transfer to Pre-Discharge Unit/Rehab	PDUTRANSDATE	Date	Date when a patient is transferred to a registered PreDischarge Unit/Rehab within the hospital prior to discharge.
Discharge Code	DISC_CODE	Integer	Identifies the discharge destination.
Discharge Status	DIS_STATUS	Integer	Public/Private status of patient on discharge.
Discharge Ward	DISWARD	Alphanumeric	The ward occupied prior to discharge.
Hospital Number	HOSPITAL	Alphanumeric	Unique numeric identifier assigned to each hospital by HIPE for operational purposes.
Medical Record Number	MRN	Alphanumeric	Unique identifier assigned by hospital to this patient within this hospital but not necessarily unique to this patient encounter. All encounters relating to a particular patient in the same hospital should be recorded under the same MRN, but this is not always possible.
Mode of Emergency Admission	EMADM	Integer	Indicates where the patient was treated prior to being admitted into the hospital as an emergency in-patient or when the patient was treated only in a registered Acute Medical Assessment Unit (AMAU). (AMAU must be registered with Healthcare Pricing Office).
Non Acute Length of Stay	TLDAYS	Integer	Number of days a patient did not spend in hospital during an episode.
Number of Days in Critical Care Bed	CriticalCareBedDays	Integer	This identifies the number of days a patient spent in a Level 2, Level 3 or Level 3s critical care bed.
Principal Diagnosis	DIAG1	Alphanumeric	ICD-10 diagnosis established after study to be chiefly responsible for occasioning the episode of admitted patient care.
Principal Procedure	PROC1	Integer	The procedure performed for treatment of the principal diagnosis or a diagnostic/exploratory procedure related to the principal diagnosis.

Table: Dataset Variables (cont'd).

Variable	Variable Name in Database	Data Type	Description
Private Bed	PRIVDAYS	Integer	Number of days patient spent in a private bed.
Public Bed	PUBDAYS	Integer	Number of days patient spent in a public bed.
Semi Private Bed	SEMIPRIVDAYS	Integer	Number of days patient spent in a semi private bed.
Sex	SEX	Integer	The sex of the patient.
Age	AGE	Integer	The age of patient.
Source of Admission	ADMSOURCE	Integer	Describes where the patient was admitted from. It does not refer to where an emergency or accident occurred.
Time of Admission	ADMTIME	Numeric	Time patient is admitted to ward.
Time of Discharge	DISTIME	Numeric	Time patient is discharged from hospital.
Transfer Hospital: From	TransInHosp	Integer	Hospital from which the patient was transferred.
Transfer Hospital: To	TransOutHosp	Integer	Hospital to which the patient was transferred.
Type of Admission	ADMTYPE	Integer	The category of admission relating to this episode of care, indicates the priority of the admission.
Fracture Type	ADMFRACTURETYPE	Integer	Type of hip fracture.
Admission Fragility	ADMFRAGILITY	Boolean	Indicates the fragility history of patients.
Length of Stay	LOS	Integer	The inpatient length of stay.

Appendix II: Public Hospitals Participating in IHFD Repository

Table: List of hospitals.

Hospital Name	County
AMNCH Tallaght Hospital	Dublin
Beaumont Hospital	Dublin
Connolly Hospital Blanchardstown	Dublin
Cork University Hospital	Cork
Galway University Hospitals	Galway
University Hospital Kerry	Kerry
Letterkenny University Hospital	Donegal
Mater Misericordiae University Hospital	Dublin
Mayo University Hospital	Mayo
Midland Regional Hospital, Tullamore	Offaly
Our Lady of Lourdes Hospital, Drogheda	Louth
Sligo University Hospital	Sligo
St. James's Hospital	Dublin
St. Vincent's University Hospital	Dublin
University Hospital Limerick	Limerick
University Hospital Waterford	Waterford

Appendix III: Code Snippets

```
library(ggplot2)
data <- read.csv("Clusters.csv")
wss <- (nrow(data)-1)*sum(apply(data,2,var))

# WSS for K=2 to 7
for (i in 2:7){
    wss[i] <- sum(kmeans(data, centers=i)$withinss)
  }
  clusterCenters <- data.frame(K=1:7, WSS=wss)

  ggplot(data=clusterCenters, aes(x=K, y=WSS)) +
  geom_line(linetype = "dashed",size=1,color="gray40")+
  geom_point(size=4, color="gray0") +
  xlab("Number of Clusters (K)")+
  ylab("Within Clusters Sum of Squares")
```

Figure: R-Script used for plotting clusters WSS.

```
library(arules)
library(arulesViz)
data<-read.csv("IHFDDataset.csv")

#Converting columns into factor-based values
data$Hosp <-as.factor(data$Hosp)
data$CHO <-as.factor(data$CHO)
data$Gender <-as.factor(data$Gender)
data$Resid <-as.factor(data$Resid)
data$Diag <-as.factor(data$Diag)
data$FracType <-as.factor(data$FracType)
data$Fragility <-as.factor(data$Fragility)
data$TTS <-as.factor(data$TTS)
data$Age <-as.factor(data$Age)
data$Cluster <-as.factor(data$Cluster)
data$LOS <-as.factor(data$LOS)

rules <- apriori(data,
                 control = list(verbose=F),
                 parameter = list( supp=0.1, conf=0.80),
                 appearance = list(rhs=c("LOS=Above-Avg",
                                         "LOS=Avg"),
                                   default="lhs"))
quality(rules) <- round(quality(rules), digits=3)
rules.sorted <- sort(rules, by="confidence")
inspect(rules.sorted)

#Plotting rules
plot(rules.sorted, method="graph", control=list(type="items"))
```

Figure: R-Script used for rule mining.

```
library(deSolve)
START<-2017;FINISH=2027;STEP<-1 #Initialise simulation start and
finish times, and time step

simTime <- seq(START, FINISH, by=STEP) #Create a simulation time
vector, needed by deSolve

malePop <-  c(
437907.0,453122.0,468454.0,484698.0,501711.0,518177.0,535249.0,552
724.0,570513.0,588297.0,611828.0)

femalePop <-
c(479079.0,493389.0,507616.0,523315.0,539759.0,555410.0,571876.0,5
88662.0,606098.0,622986.0,641675.0)

popLookup<-
data.frame(Time=simTime,MalePopulation=malePop,FemalePopulation=fe
malePop)

patientClusters<-c("Cluster1","Cluster2","Cluster3")
cluster1Prop = (2508/5504)
cluster2Prop = (984/5504)
cluster3Prop = (2012/5504)
popPercent<- c(cluster1Prop,cluster2Prop,cluster3Prop)

clusterLookup <-
data.frame(Cluster=patientClusters,Population=popPercent)

model<- function(time,stocks,auxs)
{
  with(as.list(c(stocks,auxs)),{
    #Calculate the flows
    #Looking up the corrsponding inflow and outflow at the current
simulation time
    df<- popLookup[popLookup$Time==time,]
    fMalePatients<-(140.0 / 100000.0 ) * df$MalePopulation
    fFemalePatients<- (407.0 / 100000.0) * df$FemalePopulation

    sDiscahrgedPatients <- round ( (fMalePatients+
fFemalePatients),0 )
    #Cluster1 Patients flow
    cluster1Percent<-
(clusterLookup[clusterLookup$Cluster=="Cluster1",]$Population )
    fcluster1_Males<-round((cluster1Percent*fMalePatients),0)
    fcluster1_Females<-round((cluster1Percent*fFemalePatients),0)
    #Cluster2 Patients flow
    cluster2Percent<-
(clusterLookup[clusterLookup$Cluster=="Cluster2",]$Population )
    fcluster2_Males<-round((cluster2Percent*fMalePatients),0)
    fcluster2_Females<-round((cluster2Percent*fFemalePatients),0)
    #Cluster3 Patients flow
    cluster3Percent<-
(clusterLookup[clusterLookup$Cluster=="Cluster3",]$Population )
    fcluster3_Males<-round((cluster3Percent*fMalePatients),0)
```

```
    fcluster3_Females<-round((cluster3Percent*fFemalePatients),0)

    #return calculations as a list,
    return (list(c(sDiscahrgedPatients),
                  Cluster1_Males=fcluster1_Males,
Cluster1_Females=fcluster1_Females,
                  Cluster2_Males=fcluster2_Males,
Cluster2_Females=fcluster2_Females,
                  Cluster3_Males=fcluster3_Males,
Cluster3_Females=fcluster3_Females
    ))
})
 }

#Create the vector of stocks, and its initial value
stocks<-c(sDiscahrgedPatients=0)

#Run simulation
simOutput<- data.frame(ode(y=stocks,simTime,func = model,parms =
NULL,method = "euler"))

write.csv(x = simOutput,"SDModelProjections.csv") #simOutput
```

Figure: R-Script used for the SD model.

```
library ("RCurl")
library ("rjson")

generatedPatients<- read.csv("Experiments.csv")
# Accept SSL certificates issued by public Certificate Authorities
options(RCurlOptions = list(cainfo = system.file("CurlSSL", "cacert.pem",
package = "RCurl")))

h = basicTextGatherer()
hdr = basicHeaderGatherer()
predictions <-  data.frame()
StepSize <-1000
for( i in seq(1,nrow(generatedPatients), by=StepSize)){
  partition<- list(0)
  counter <- i
  for(j in 1:StepSize)
  {
    partition[[j]]<-list( generatedPatients[counter,"Hosp"],
                          generatedPatients[counter,"Sex"],
                          generatedPatients[counter,"Resid"],
                          generatedPatients[counter,"AGE"],
                          generatedPatients[counter,"Diag1"],
                          generatedPatients[counter,"Fracture_Type"],
                          generatedPatients[counter,"Fragility"],
                          "0",
                          generatedPatients[counter,"TimeToSurgery"],
                          generatedPatients[counter,"TimeToAdmission"] )
    counter<- counter+1
  }

  req = list(
    Inputs = list(
      "input1" = list(
        "ColumnNames" = list("Hosp", "Sex", "Resid", "AGE", "Diag1",
"Fracture_Type", "Fragility", "ActualLOS", "TimeToSurgery", "TimeToAdmission"),
        "Values" = partition
      )                 ),
    GlobalParameters = setNames(fromJSON('{}'), character(0))
  )

  body = enc2utf8(toJSON(req))
  api_key = ""  # Replace this with the API key for the web service
  authz_hdr = paste('Bearer', api_key, sep=' ')
  h$reset()
  curlPerform(url =
"https://ussouthcentral.services.azureml.net/workspaces/ee457a75262244d4b299080febd209db/services
/d7cd2674a7e046e685b1ad12ad1f79b3/execute?api-version=2.0&details=true",
              httpheader=c('Content-Type' = "application/json", 'Authorization' =
authz_hdr),
              postfields=body,
              writefunction = h$update,
              headerfunction = hdr$update,
              verbose = FALSE
  )
```

```
  headers = hdr$value()
  httpStatus = headers["status"]
  if (httpStatus >= 400)
  {
    print(paste("The request failed with status code:", httpStatus, sep=" "))
    # Print the headers - they include the requert ID and the timestamp, which are useful for
debugging the failure
    print(headers)
  }
    result = h$value()
  counter <- i
  for(j in 1:StepSize)
  {
    predictedLOS <-
fromJSON(result)$Results$output1$value$Values[[j]][11]
    #Denormalisation
    val <- round( as.numeric(predictedLOS)*(60-1)+1, 0)
    #print(val)
    predictions[counter,"PredictedLOS"]<-val
    counter <- counter+1
  }
}
write.csv(predictions,"Predictions.csv")
```

Figure: R-Script used for calling the Azure predictive web service used to predict LOS.

Bibliography

A. Newell, & H.A. Simon. (1959). *The Simulation of Human Thought*. Current Trends in Psychological Theory. University of Pittsburgh Press.

Abu-Mostafa, Y.S., Magdon-Ismail, M. & Lin, H.T. (2012). *Learning from Data* (Vol. 4). Singapore: AMLBook.

Agrawal, R., & Srikant, R. (1994, September). Fast Algorithms for Mining Association Rules. In Proceedings of 20th International Conference on Very Large Databases, VLDB (Vol. 1215, pp. 487-499).

Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining Association Rules between Sets of Items in Large Databases. In ACM SIGMOD record (Vol. 22, No. 2, pp. 207-216). ACM.

Ahmed, F., Robinson, S., & Tako, A. A. (2014). Using the Structured Analysis and Design Technique (SADT) in Simulation Conceptual Modeling. In Proceedings of the 2014 Winter Simulation Conference (WSC), (pp. 1038-1049). IEEE.

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation Importance: A Corrected Feature Importance Measure. Bioinformatics, 26(10), 1340-1347.

Arbez, G. ,& Birta, L. G. 2011. The ABCmod Conceptual Modeling Framework. In Conceptual Modeling for Discrete-Event Simulation, edited by S. Robinson, S., R. J. Brooks, K. Kotiadis, and D-J. van der Zee, 133-178. Boca Raton, FL: Chapman and Hall/CRC.

Azevedo, A. and Santos, M. F. (2008). KDD, SEMMA and CRISP-DM: A Parallel Overview. In Proceedings of the IADIS European Conference on Data Mining.

Azhar, A., Lim, C., Kelly, E., O'Rourke, K., Dudeney, S., Hurson, B., & Quinlan, W. (2008). Cost Induced by Hip Fractures. Irish Medical Journal, 101(7), 213-215.

Azure ML. (2017). https://gallery.cortanaintelligence.com/Experiment/Patient-Clustering-2013-2015.

Barga, R., Fontama, V., Tok, W. H., & Cabrera-Cordon, L. (2015). *Predictive Analytics with Microsoft Azure Machine Learning*. Apress.

Bayardo, R. J., Agrawal, R., & Gunopulos, D. (1999). Constraint-Based Rule Mining in Large, Dense Databases. In Proceedings of 15th International Conference on Data Engineering. IEEE.

Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J. F., & Gandrillon, O. (2002). Strong-Association-Rule Mining for Large-Scale Gene-Expression Data Analysis: A Case Study on Human SAGE Data. Genome Biology, 3(12), Research0067-1.

Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press.

Bergmann, S., & Stelzer, S. (2011, June). Approximation of Dispatching Rules in Manufacturing Control Using Artificial Neural Networks. In Principles of Advanced and Distributed Simulation (PADS), 2011 IEEE Workshop on (pp. 1-8). IEEE.

Blum, A. L., & Langley, P. (1997). Selection of Relevant Features and Examples in Machine Learning. Artificial Intelligence, 97(1), 245-271.

Borshchev, A. (2013). T*he Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6*. AnyLogic North America.

Bradley, P. S., & Fayyad, U. M. (1998, July). Refining Initial Points for K-Means Clustering. In ICML (Vol. 98, pp. 91-99).

Brailsford, S. C., and Hilton, N. A. (2001). A Comparison of Discrete Event Simulation and System Dynamics for Modelling Health Care Systems. In Proceedings of ORAHS, Glasgow, Scotland, pp 18-39.

Breiman, L. (2001). Random Forests. Machine learning, 45(1), 5-32.

British Orthopaedic Association. (2007). The Care of Patients with Fragility Fracture. London: British Orthopaedic Association, pp.8-11.

Brossette, S. E., Sprague, A. P., Hardin, J. M., Waites, K. B., Jones, W. T., & Moser, S. A. (1998). Association Rules and Data Mining in Hospital Infection Control and Public Health Surveillance. Journal of the American Medical Informatics Association, 5(4), 373-381.

Carey, D., & Laffoy, M. (2005). Hospitalisations due to Falls in Older Persons. Irish Medical Journal, 98(6), 179-181.

Caruana, R., & Sa, V. R. D. (2003). Benefitting from the Variables that Variable Selection Discards. Journal of Machine Learning Research, 3(Mar), 1245-1264.

Ceglowski, R., Churilov, L., & Wasserthiel, J. (2007). Combining Data Mining and Discrete Event Simulation for a Value-Added View of a Hospital Emergency Department. Journal of the Operational Research Society, 58(2), 246-254.

Cellier, F. E. (1991). Qualitative Modeling and Simulation: Promise or Illusion. In Proceedings of the 23rd Conference on Winter Simulation (pp. 1086-1090). IEEE Computer Society.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357.

Chizi, B., & Maimon, O. (2009). Dimension Reduction and Feature Selection. In Data mining and Knowledge Discovery Handbook (pp. 83-100). Springer US.

Cleary, J., Goh, K. S., and Unger, B. W. (1985). Discrete Event Simulation in Prolog. Proc. of Artificial Intelligence, Graphics and Simulation.

Cohen, K. J., and Cyert, R. M. (1965). *Simulation of Organizational Behavior*. *Handbook of Organizations* (305-334). Rand McNally, Chicago.

Cooley, R., Mobasher, B., & Srivastava, J. (1997, November). Web Mining: Information and Pattern Discovery on the World Wide Web. In Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on (pp. 558-567). IEEE.

Cooper, C., Campion, G., & Melton, L. 3. (1992). Hip Fractures in the Elderly: A World-Wide Projection. Osteoporosis International, 2(6), 285-289.

Cotter, P. E., Timmons, S., O'Connor, M., Twomey, C., & O'Mahony, D. (2006). The Financial Implications of Falls in Older People for an Acute Hospital. Irish Journal of Medical Science, 175(2), 11-13.

Creighton, C., & Hanash, S. (2003). Mining Gene Expression Databases for Association Rules. Bioinformatics, 19(1), 79-86.

CSO. (2017). Retrieved from http://www.cso.ie/en/statistics/population/.

Davenport, T. H., & Harris, J. G. (2007). *Competing on Analytics: The New Science of Winning.* Harvard Business Press.

Djanatliev, A., & German, R. (2013, December). Prospective Healthcare Decision-Making by Combined System Dynamics, Discrete-Event and Agent-Based Simulation. In Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World (pp. 270-281). IEEE Press.

DESMO-J Tutorial. (2017). Retrieved from http://desmoj.sourceforge.net/tutorial/nutshell/1.html.

Dodds, M. K., Codd, M. B., Looney, A., & Mulhall, K. J. (2009). Incidence of hip fracture in the Republic of Ireland and Future Projections: A Population-Based Study. Osteoporosis International, 20(12), 2105-2110.

Domingos, P. (2012). A Few Useful Things to Know about Machine Learning. Communications of the ACM, 55(10), 78-87.

Duggan, J. (2015). Functions, DataFrames and deSolve [CT561-Lecture slides].

Duggan, J. (2016). *System Dynamics Modeling with R.* Springer.

Elbattah. (2017). Retrieved from https://github.com/Mahmoud-Elbattah/DESModel_SIGSIM2016.

El-Darzi, E., Abbi, R., Vasilakis, C., Gorunescu, F., Gorunescu, M., & Millard, P. (2009). Length of Stay-Based Clustering Methods for Patient Grouping. Intelligent Patient Management, 39-56.

Ellanti, P., Cushen, B., Galbraith, A., Brent, L., Hurson, C., & Ahern, E. (2014). Improving Hip Fracture Care in Ireland: A Preliminary Report of the Irish Hip Fracture Database. Journal of Osteoporosis, 2014.

Englert, J., Davis, K. M., & Koch, K. E. (2001). Using Clinical Practice Analysis to Improve Care. The Joint Commission Journal on Quality Improvement, 27(6), 291-301.

Estivill-Castro, V., & Yang, J. (2000). Fast and Robust General Purpose Clustering Algorithms. In Pacific Rim International Conference on Artificial Intelligence (pp. 208-218). Springer Berlin Heidelberg.

European Commission. 2009. The 2009 Ageing Report: Underlying Assumptions and Projection Methodologies for the EU-27 Member States (2007-2060) (No. 7). Office for Official Publications of the European Communities.

Faddy, M. J., & McClean, S. I. (1999). Analysing Data on Lengths of Stay of Hospital Patients Using Phase-type Distributions. Applied Stochastic Models in Business and Industry, 15(4), 311-317.

Feldkamp, N., Bergmann, S., & Strassburger, S. (2015, June). Knowledge Discovery in Manufacturing Simulations. In Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (pp. 3-12). ACM.

Feldkamp, N., Bergmann, S., Strassburger, S., & Schulze, T. (2016, December). Knowledge Discovery in Simulation Data: A Case Study of a Gold Mining Facility. In Winter Simulation Conference (WSC), 2016 (pp. 1607-1618). IEEE.

Fisher, R. A. (1938). The Statistical Utilization of Multiple Measurements. Annals of Human Genetics, 8(4), 376-386.

Fishwick, P. A. (1995a). Computer Simulation: The Art and Science of Digital World Construction. Department of Computer and Information Science and Engineering, University of Florida.

Fishwick, P. A. (1995b). *Simulation Model Design and Execution: Building Digital Worlds*. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Fishwick, P. A., & Modjeski, R.B. (2012). *Knowledge-Based Simulation: Methodology and Application* (Vol. 4). Springer Science & Business Media.

Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research, 3(Mar), 1289-1305.

Forrester, J. W. (1994). System Dynamics, Systems Thinking, and soft OR. System Dynamics Review, 10(2-3), 245-256.

Forrester, J.W. (1960). The Impact of Feedback Control Concepts on the Management Sciences. Foundation for Instrumentation Education and Research.

Forrester, J.W. (1961). *Industrial Dynamics*. MIT Press, Cambridge, MA.

Forrester, J.W. (1964). Common Foundations Underlying Engineering and Management. IEEE spectrum, 1(9), pp.66-77.

Forrester, J.W. (1968). *Principles of Systems*. MIT Press, Cambridge, MA.

Forrester, J.W. (1971). Counterintuitive Behavior of Social Systems. Technological Forecasting and Social Change, 3, 1-22.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press.

Futó, I., and Gergely, T. (1986). TS-PROLOG, a Logic Simulation Language. Transactions of the Society for Computer Simulation International, 3(4), 319-336.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(4), 463-484.

Gannon, B., O'Shea, E., & Hudson, E. (2007). The Economic Costs of Falls and Fractures in People Aged 65 and over In Ireland. Irish Centre for Social Gerontology, Galway.

Garrido, J. M. (2012). Object-Oriented Discrete-Event Simulation with Java: A Practical Introduction. Springer Science & Business Media.

Gold, J. (2015). Accountable Care Organizations, Explained. Kaiser Health News.

Gröne, O., & Garcia-Barbero, M. (2001). Integrated Care. International Journal of Integrated Care, 1(2).

Gullberg, B., Johnell, O., & Kanis, J. A. (1997). Worldwide Projections for Hip Fracture. Osteoporosis international, 7(5), 407-413.

Guizzardi, G., & Wagner, G. (2012). Conceptual Simulation Modeling with Onto-UML. In Proceedings of the 2012 Winter Simulation Conference (p. 5). ACM.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3(Mar), 1157-1182.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3(Mar), 1157-1182.

Karagöz, N. A., & Demirörs, O. (2011). Conceptual Modeling Notations and Techniques. In Conceptual Modeling for Discrete-Event Simulation, edited by S. Robinson, S., R. J. Brooks, K. Kotiadis, and D-J. van der Zee, 179-209. Boca Raton, FL: Chapman and Hall/CRC.

Lee, Y. H., Cho, M. K., Kim, S. J., & Kim, Y. B. (2002). Supply Chain Simulation with Discrete–Continuous Combined Modeling. Computers & Industrial Engineering, 43(1-2), 375-392.

Haentjens, P., Lamraski, G., & Boonen, S. (2005). Costs and Consequences of Hip Fracture Occurrence in Old Age: An Economic Perspective. Disability and Rehabilitation, 27(18-19), 1129-1141.

Hahsler, M., & Chelluboina, S. (2011). Visualizing Association Rules: Introduction to the R-Extension Package arulesViz. R Project Module, 223-238.

Hahsler, M., Grün, B., & Hornik, K. (2005). A Computational Environment for Mining Association Rules and Frequent Item Sets. Journal of Statistical Software 14(15), 1–25.

Halkidi, M., & Vazirgiannis, M. (2009). Quality Assessment Approaches in Data Mining. In Data Mining and Knowledge Discovery Handbook (pp. 613-639). Springer US.

Harper, P. R., and Shahani, A. K. (2002). Modelling for the Planning and Management of Bed Capacities in Hospitals. Journal of the Operational Research Society, 53(1), 11-18.

Harris, M.D. (1991). Clinical and Financial Outcomes in Patient care in a Home Health Care Agency. Journal of Nursing Care Quality, 5(2), pp.41-49.

Higgins, J. R. (2013). The Establishment of Hospital Groups as a Transition to Independent Hospital Trusts. Report to the Minister of Health. Department of Health, 2013.

HIPE. (2015). Retrieved from http://www.hpo.ie/hipe/hipe_data_dictionary/HIPE_Data_Dictionary_2015_V7.0.pdf.

HIQA. (2016). Retrieved from https://www.hiqa.ie/social-care/find-a-centre/nursing-homes.

Hodge, V.J. & Austin, J., (2004). A Survey of Outlier Detection Methodologies. Artificial Intelligence Review, 22(2), pp.85-126.

HSE. (2008). Retrieved from http://www.hse.ie/eng/services/publications/olderpeople/Executive_Summary_-_Strategy_to_Prevent_Falls_and_Fractures_in_Ireland%E2%80%99s_Ageing_Population.pdf

HSE. (2014). Annual Report and Financial Statements 2014.

HSE. (2015). Retrieved from http://www.hse.ie/eng/services/publications/corporate/CHO_Chapter_1.pdf.

Hu, M., & Liu, B. (2004, July). Mining Opinion Features in Customer Reviews. In AAAI (Vol. 4, No. 4, pp. 755-760).

Huang, Y. (2013). Automated Simulation Model Generation. (Doctoral dissertation, TU Delft, Delft University of Technology).

Huang, Z. (1998). Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data mining and knowledge discovery, 2(3), 283-304.

Hwang, J. N., Lay, S. R., & Lippman, A. (1994). Nonparametric Multivariate Density Estimation: A Comparative Study. IEEE Transactions on Signal Processing, 42(10), 2795-2810.

Ihaka, R. (1998). R: Past and Future History. Computing Science and Statistics, 392396.

Ihaka, R., & Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. Journal of Computational and Graphical Statistics, 5(3), 299-314.

Isken, M. W., & Rajagopalan, B. (2002). Data Mining to Support Simulation Modeling of Patient Flow in Hospitals. Journal of medical systems, 26(2), 179-197.

Jackson, J. (1991). *A User's Guide to Principal Components*. New York: John Wiley and Sons.

Jain, A. K. (2010). Data Clustering: 50 Years beyond K-Means. Pattern Recognition Letters, 31(8), 651-666.

Japkowicz, N., & Stephen, S. (2002). The Class Imbalance Problem: A Systematic Study. Intelligent Data Analysis, 6(5), 429-449.

Jimenez, L. O., & Landgrebe, D. A. (1998). Supervised Classification in High-Dimensional Space: Geometrical, Statistical, and Asymptotical Properties of Multivariate Data. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 28(1), 39-54.

Johansen, A., Wakeman, R., Boulton, C., Plant, F., Roberts, J. & Williams, A. (2013). National Hip Fracture Database: National Report 2013. Clinical Effectiveness and Evaluation Unit at the Royal College of Physicians.

Jolliffe, I. (1986). *Principal Component Analysis*. Springer-Verlag.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research, 4, 237-285.

Kannampallil, T. G., Schauer, G. F., Cohen, T., & Patel, V. L. (2011). Considering Complexity in Healthcare Systems. Journal of Biomedical Informatics, 44(6), 943-947.

Kannus, P., Parkkari, J., Sievänen, H., Heinonen, A., Vuori, I., & Järvinen, M. (1996). Epidemiology of hip fractures. Bone, 18(1), S57-S63.

Karres, J., Heesakkers, N. A., Ultee, J. M., & Vrouenraets, B. C. (2015). Predicting 30-Day Mortality Following Hip Fracture Surgery: Evaluation of Six Risk Prediction Models. Injury, 46(2), 371-377.

Katsaliaki, K., & Mustafee, N. (2011). Applications of Simulation within the Healthcare Context. Journal of the Operational Research Society, 62(8), 1431-1451.

Kuipers, B. (1986). Qualitative Simulation. Artificial Intelligence, 29(3), pp.289-338.

Laffoy, M. (2008). Strategy to Prevent Falls and Fractures in Ireland's Ageing Population Summary, Conclusions and Recommendations.

Laite, R., Portman, N., & Sankaranarayanan, K. (2016, December). Behavioral Analysis of Agent Based Service Channel Design Using Neural Networks. In Proceedings of the 2016 Winter Simulation Conference (pp. 3694-3695). IEEE Press.

Lane, D.C. (2000).You Just Don't Understand Me: Modes of Failure and Success in the Discourse between System Dynamics and Discrete Event Simulation. LSE OR Dept Working Paper LSEOR 00-34, London School of Economics and Political Science.

Lattner, A. D., Bogon, T., Lorion, Y., and Timm, I. J. (2010). A Knowledge-Based Approach to Automated Simulation Model Adaptation. In Proceedings of the 2010 Spring Simulation Multiconference (p. 153). Society for Computer Simulation International.

Laursen, G. H., & Thorlund, J. (2016). B*usiness Analytics for Managers: Taking Business Intelligence beyond Reporting*. John Wiley & Sons.

Law, A. M. (2007). *Simulation Modeling & Analysis.* 4th ed. Singapore: McGraw-Hill Education.

Law, A. M. (2008). How to Build Valid and Credible Simulation Models. In Proceedings of the 40th Conference on Winter Simulation (pp. 39-47). IEEE.

Lechler, T. & Page, B., (1999). DESMO-J: An Object-Oriented Discrete Simulation Framework in Java. In Proceedings of the 11th European Simulation Symposium (pp. 46-50).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436-444.

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of Massive Datasets*. Cambridge University Press.

Liberatore, M., & Luo, W. (2011). INFORMS and the Analytics Movement: The View of the Membership. Interfaces, 41(6), 578-589.

Lutz, W., Sanderson, W., & Scherbov, S. (2008). The Coming Acceleration of Global Population Ageing. Nature, 451(7179), 716-719.

Marshall, A. H., & McClean, S. I. (2003). Conditional Phase-Type Distributions for Modelling Patient Length of Stay in Hospital. International Transactions in Operational Research, 10(6), 565-576.

Martínez, A. M., & Kak, A. C. (2001). PCA versus lDA. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2), 228-233.

Martis, M. S. (2006). Validation of Simulation Based Models: A Theoretical Outlook. The Electronic Journal of Business Research Methods, 4(1), 39-46.

Marufu, T. C., White, S. M., Griffiths, R., Moonesinghe, S. R., & Moppett, I. K. (2016). Prediction of 30-Day Mortality After Hip Fracture Surgery by the Nottingham Hip Fracture Score and the Surgical Outcome Risk Tool. Anaesthesia, 71(5), 515-521.

McGowan, B., Casey, M. C., Silke, C., Whelan, B., & Bennett, K. (2013). Hospitalisations for Fracture and Associated Costs between 2000 and 2009 in Ireland: A Trend Analysis. Osteoporosis International, 24(3), 849-857.

Meadows, D. H. (2008). *Thinking in Systems: A Primer*. Chelsea Green Publishing.

Melton, L. (1993). Hip Fractures: A Worldwide Problem Today and Tomorrow. Bone, 14, 1-8.

Melton, L. J. (1996). Epidemiology of Hip Fractures: Implications of the Exponential Increase with Age. Bone, 18(3), S121-S125.

Miller, D. P., Firby, R. J., Fishwick, P. A., & Rothenberg, J. (1992). AI: What Simulationists Really Need to Know. ACM Transactions on Modeling and Computer Simulation (TOMACS), 2(4), 269-284.

Mitchell, T.M. (1997). *Machine Learning*. 1997. Burr Ridge, IL: McGraw Hill, 45, p.37.

Monostori, L., Kádár, B., Viharosy, Z. J., & Stefan, P. (2000). Combined Use of Simulation and AI/Machine Learning Techniques In Designing and Manufacturing Processes and Systems. Journal for Manufacturing Science and Production, 3(2-4), 111-118.

Morin, M., Paradis, F., Rolland, A., Wery, J., Gaudreault, J., & Laviolette, F. (2015, December). Machine learning-based metamodels for sawing simulation. In Proceedings of the 2015 Winter Simulation Conference (pp. 2160-2171). IEEE Press.

Mortenson, M. J., Doherty, N. F., & Robinson, S. (2015). Operational Research from Taylorism to Terabytes: A Research Agenda for the Analytics Age. European Journal of Operational Research, 241(3), 583-595.

Mustafee, N., Brailsford, S., Djanatliev, A., Eldabi, T., Kunc, M., Tolk, A. (2017, December). Purpose and Benefits of Hybrid Simulation: Contributing to the Convergence of its Definition. In Proceedings of the 2017 Winter Simulation Conference (pp. 1631-1645). IEEE Press.

Nahar, J., Imam, T., Tickle, K. S., & Chen, Y. P. P. (2013). Association Rule Mining to Detect Factors Which Contribute to Heart Disease in Males and Females. Expert Systems with Applications, 40(4), 1086-1093.

Newcomer, S. R., Steiner, J. F., & Bayliss, E. A. (2011). Identifying Subgroups of Complex Patients with Cluster Analysis. The American Journal of Managed Care, 17(8), e324-32.

Ng, H. P., Ong, S. H., Foong, K. W. C., Goh, P. S., & Nowinski, W. L. (2006, March). Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm. In proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation, (pp. 61-65). IEEE.

Nijmeijer, W. S., Folbert, E. C., Vermeer, M., Slaets, J. P., & Hegeman, J. H. (2016). Prediction of Early Mortality Following Hip Fracture Surgery in Frail Elderly: The Almelo Hip Fracture Score (AHFS). Injury, 47(10), 2138-2143.

NOCA. (2015). Retrieved from https://www.noca.ie/wp-content/uploads/2015/04/NOCA-IHFD-National-Report-2015-FINAL.pdf.

NOCA. (2017). Retrieved from https://www.noca.ie/irish-hip-fracture-database.

Nolan, M.T. & Mock, V., 2000. *Measuring Patient Outcomes*. Sage Publications.

Oberkampf, W. L., DeLand, S. M., Rutherford, B. M., Diegert, K. V., and Alvin, K. F. (2002). Error and Uncertainty in Modeling and Simulation. Reliability Engineering & System Safety, 75(3), 333-357.

O'keefe, G. E., Jurkovich, G. J., & Maier, R. V. (1999). Defining Excess Resource Utilization and Identifying Associated Factors for Trauma Victims. Journal of Trauma and Acute Care Surgery, 46(3), 473-478.

Olson, D.L. and Delen, D. (2008). *Advanced Data Mining Techniques*. Springer Science & Business Media.

Page, B., & Kreutzer, W. (2005). *The Java Simulation Handbook: Simulating Discrete Event Systems with UML and Java*. Shaker.

Painter, M. K., Erraguntla, M., Hogg, G. L., & Beachkofski, B. (2006, December). Using Simulation, Data Mining, and Knowledge Discovery Techniques for Optimized Aircraft Engine Fleet Management. In Proceedings of the 2006 Winter Simulation Conference, (pp. 1253-1260). IEEE.

Parker, M., & Johansen, A. (2006). Hip fracture. BMJ: British Medical Journal, 333(7557), 27.

Patel, V. R., & Mehta, R. G. (2011). Impact of Outlier Removal and Normalization Approach in Modified K-Means Clustering Algorithm. IJCSI International Journal of Computer Science Issues, 8(5), 331-336.

Peng, R. D. (2015). *R Programming for Data Science*. Lulu. com.

Pidd, M. (2003). *Tools for Thinking: Modelling in Management Science*, 2nd ed. Wiley, Chichester, UK.

Pidd, M. (2004). *Systems Modelling: Theory and Practice*. John Wiley & Sons: Chichester, England.

Powell, J., & Mustafee, N. (2014). Soft OR Approaches in Problem Formulation Stage of a Hybrid M&S Study. In Proceedings of the 2014 Winter Simulation Conference (pp. 1664-1675). IEEE Press.

Powell, J. H., & Mustafee, N. (2017). Widening Requirements Capture with Soft Methods: An Investigation of Hybrid M&S Studies in Health Care. Journal of the Operational Research Society, 68(10), 1211-1222.

Pruyt, E. (2017). Integrating Systems Modelling and Data Science: The Joint Future of Simulation and 'Big Data' Science. In Artificial Intelligence: Concepts, Methodologies, Tools, and Applications (822-840). IGI Global.

Pruyt, E., Cunningham, S., Kwakkel, J.H. and De Bruijn, J.A., 2014. From Data-Poor to Data-Rich: System Dynamics in the Era of Big Data. In Proceedings of the 2014 International Conference of the System Dynamics Society, System Dynamics Society. Delft, Netherlands.

Rabelo, L., Cruz, L., Bhide, S., Joledo, O., Pastrana, J., & Xanthopoulos, P. (2014). Analysis of the Expansion of the Panama Canal Using Simulation Modeling and Artificial Intelligence. In Proceedings of the 2014 Winter Simulation Conference (pp. 910-921). IEEE Press.

Ragab, M., Abo-Hamad, W., & Arisha, A., (2012). Capacity Planning for Elderly Care in Ireland Using Simulation Modeling. In Proceedings of the International Conference on Advances in System Simulation.

Rand, W. (2006, September). Machine Learning Meets Agent-Based Modeling: When not to go to a bar. In Proceedings of the Conference on Social Agents: Results and Prospects. (pp. 51-59). Chicago, IL: Argonne National Laboratory-University of Chicago.

Rashwan, W., Ragab, M., Abo-Hamad, W., & Arisha, A. (2013). Evaluating Policy Interventions for Delayed Discharge: A System Dynamics Approach. In Proceedings of the 2013 Winter Simulation Conference: (pp. 2463-2474). IEEE Press.

Rechel, B., Grundy, E., Robine, J. M., Cylus, J., Mackenbach, J. P., Knai, C., & McKee, M. (2013). Ageing in the European Union. The Lancet, 381(9874), 1312-1322.

Richardson, G. (1983). The Feedback Concept in American Social Science, with Implications for System Dynamics. In Proceedings of the 1983 International System Dynamics Conference, Massachusetts.

Robinson, S. (2004). *Simulation: The Practice of Model Development and Use*. Chichester: Wiley.

Robinson, S. (2011, December). Choosing the Right Model: Conceptual Modeling for Simulation. In Proceedings of the 2011 Winter Simulation Conference (WSC), (pp. 1423-1435). IEEE.

Robinson, S. (2015, December). A Tutorial on Conceptual Modeling for Simulation. In Proceedings of the 2015 Winter Simulation Conference (pp. 1820-1834). IEEE Press.

Roger, D.P. (2015). *Exploratory Data Analysis with R*. Lean Publishing.

Rokach, L. (2009). A Survey of Clustering Algorithms. In Data Mining and Knowledge Discovery Handbook (pp.269-298). Springer US.

Rosenberg, M., & Everitt, J. (2001). Planning for Aging Populations: Inside or Outside the Walls. Progress in Planning, 56(3), 119-168.

Rouse, W. B. (2008). Health Care as a Complex Adaptive System: Implications for Design and Management. Bridge-Washington-National Academy of Engineering-, 38(1), 17.

Royston, P., Ambler, G., & Sauerbrei, W. (1999). The Use of Fractional Polynomials to Model Continuous Risk Variables in Epidemiology. International Journal of Epidemiology, 28(5), 964-974.

Sadsad, R., & McDonnell, G. (2014). Multiscale Modelling for Public Health Management: A Practical Guide. Discrete-Event Simulation and System Dynamics for Management Decision Making, 280-294.

Samuel, A.L. (1959). Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development, 3(3), pp.210-229.

Sarsfield, S., 2009. *The Data Governance Imperative*. IT Governance Publishing.

Schlimmer, J. C., & Granger, R. H. (1986). Incremental Learning from Noisy Data. Machine Learning, 1(3), 317-354.

Senge, P. (1990). *The Fifth Discipline: The Art and Science of the Learning Organization*. New York: Currency Doubleday.

Sewitch, M. J., Leffondre, K., & Dobkin, P. L. (2004). Clustering Patients According to Health Perceptions: Relationships to Psychosocial Characteristics and Medication Non-Adherence. Journal of Psychosomatic Research, 56(3), 323-332.

Shannon, R.E. (1975). *Systems Simulation: The Art and Science* (Vol. 1). Englewood Cliffs, NJ: Prentice-Hall.

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing, 5(4), 13-22.

Shreckengost, R. C. (1985). Dynamic Simulation Models: How Valid are They?. Self-Report Methods of Estimating Drug Use: Current Challenges to Validity. National Institute on Drug Abuse Research Monograph, 57, 63-70.

Simon, H. A. (1996). *The Sciences of the Artificial*. MIT press.

Simon, H.A. (1983). Why should Machines Learn?. Machine Learning (pp. 25-37). Springer Berlin Heidelberg.

Sirovich, L., & Kirby, M. (1987). Low-Dimensional Procedure for the Characterization of Human Faces. Josa a, 4(3), 519-524.

Soetaert, K. E. R., Petzoldt, T., & Setzer, R. W. (2010). Solving Differential Equations in R: Package deSolve. Journal of Statistical Software, 33.

Srinivasan, P. (2004). Text Mining: Generating Hypotheses from MEDLINE. Journal of the Association for Information Science and Technology, 55(5), 396-413.

Stanislaw, H. (1986). Tests of Computer Simulation Validity: What do they Measure?. Simulation & Games, 17(2), 173-191.

Sterman, J. D. (1994). Learning in and about Complex Systems. System Dynamics Review, 10(2-3), 291-330.

Stevenson, A. Ed. (2010). Oxford Dictionary of English. Oxford University Press, USA.

Stilou, S., Bamidis, P. D., Maglaveras, N., & Pappas, C. (2001). Mining Association Rules from Clinical Databases: An Intelligent Diagnostic Process in Healthcare. Studies in Health Technology and Informatics, (2), 1399-1403.

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of Imbalanced Data: A Review. International Journal of Pattern Recognition and Artificial Intelligence, 23(04), 687-719.

Sweetser, A. (1999). A Comparison of System Dynamics (SD) and Discrete Event Simulation (DES). In Proceedings of the 17th International Conference of the System Dynamics Society (pp. 20-23).

Tako, A. A., & Kotiadis, K. (2015). PartiSim: A Multi-Methodology Framework to Support Facilitated Simulation Modelling in Healthcare. European Journal of Operational Research, 244(2), 555-564.

Tan, J., Wen, H. J., & Awad, N. (2005). Health Care and Services Delivery Systems as Complex Adaptive Systems. Communications of the ACM, 48(5), 36-44.

Taylor, S. J., Khan, A., Morse, K. L., Tolk, A., Yilmaz, L., & Zander, J. (2013, April). Grand Challenges on the Theory of Modeling and Simulation. In Proceedings of the Symposium on Theory of Modeling & Simulation-DEVS Integrative M&S Symposium (p. 34). Society for Computer Simulation International.

Tolk, A. (2015, July). The Next Generation of Modeling & Simulation: Integrating Big Data and Deep Learning. In Proceedings of the Conference on Summer Computer Simulation (pp. 1-8). International Society for Computer Simulation.

Tolk, A., Balci, O., Combs, C. D., Fujimoto, R., Macal, C. M., Nelson, B. L., and Zimmerman, P. (2015). Do We Need a National Research Agenda for Modeling and Simulation?. In Proceedings of the 2015 Winter Simulation Conference (pp. 2571-2585). IEEE Press.

Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 3(1), 71-86.

Tyron, R.C. & Bailey, D.E. (1970). *Cluster Analysis*. McGraw-Hill.

UN. (2007). Retrieved from http://www.un.org/en/development/desa/population/pdf/commission/2007/keynote/chatterj.pdf.

UN. (2015). Retrieved from http://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2015_InfoChart.pdf.

Van der Zee, D. J. (2007). Developing Participative Simulation Models—Framing Decomposition Principles for Joint Understanding. Journal of Simulation, 1(3), 187-202.

Vijay, J., & Subhashini, J. (2013, April). An Efficient Brain Tumor Detection Methodology Using K-Means Clustering Algorithm. In Proceedings of International Conference on Communications and Signal Processing (ICCSP), (pp. 653-657). IEEE.

Visalakshi, N. K., & Thangavel, K. (2009). Impact of Normalization in Distributed K-Means Clustering. International Journal of Soft Computing, 4(4), 168-172.

Weston, J., Elisseeff, A., Schölkopf, B., & Tipping, M. (2003). Use of the Zero-Norm with Linear Models and Kernel Methods. Journal of Machine Learning Research, 3(Mar), 1439-1461.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer.

Witten, I. H., Frank, E., & Hall, M. A. (2005). Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.

Wolpert, D. H., Wheeler, K. R., & Tumer, K. (1999, April). General Principles of Learning-Based Multi-Agent Systems. In Proceedings of the third Annual Conference on Autonomous Agents (pp. 77-83). ACM.

Wolpert, D. H., & Tumer, K. *An Overview of Collective Intelligence*. In J. M. Bradshaw, editor, Handbook of Agent Technology. AAAI Press/MIT Press, 1999.

Wu, K. L., & Yang, M. S. (2002). Alternative C-Means Clustering Algorithms. Pattern Recognition, 35(10), 2267-2278.

Wu, M. N., Lin, C. C., & Chang, C. C. (2007, November). Brain Tumor Detection Using Color-Based K-Means Clustering Segmentation. In Proceedings of 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP).IEEE.

Yang, Q., & Wu, X. (2006). 10 Challenging Problems in Data Mining Research. International Journal of Information Technology & Decision Making, 5(04), 597-604.

Zeigler, B.P, & Sarjoughian, H.S. (2012). *Guide to Modeling and Simulation of Systems of Systems*. Springer Science & Business Media.

Zhong, J., Cai, W., Luo, L., & Zhao, M. (2016). Learning Behavior Patterns from Video for Agent-Based Crowd Modeling and Simulation. Autonomous Agents and Multi-Agent Systems, 30(5), 990-1019.