



An Agent-Based Reinforcement Learning Approach to Improve Human-Robot-Interaction in Manufacturing

Candidate: Harley Oliff

Student Number: C1225978

Supervisors: Dr Ying Liu, Prof Maneesh Kumar

This Thesis is submitted to Cardiff University School of Engineering in application for the award of Doctor of Philosophy (PhD).

May 2020

Table of Contents

TABLE OF CONTENTS.....	I
LIST OF FIGURES.....	III
LIST OF TABLES.....	V
ACKNOWLEDGEMENTS.....	VI
ABSTRACT	VII
1. INTRODUCTION.....	8
1.1. BACKGROUND.....	8
1.2. MAIN CHALLENGES	10
1.3. IDENTIFIED RESEARCH QUESTIONS.....	12
1.4. THESIS STRUCTURE.....	15
2. LITERATURE REVIEW	17
2.1. INTELLIGENT MANUFACTURING & INDUSTRY 4.0.....	17
2.2. DECENTRALISED CONTROL	22
2.3. MACHINE LEARNING.....	27
2.4. HUMAN-ROBOT-INTERACTION	36
2.5. HUMAN FACTORS.....	43
2.6. SUMMARY	47
3. RESEARCH FRAMEWORK & METHODS.....	52
3.1. RESEARCH FRAMEWORK	53
3.2. RESEARCH METHODS.....	56
4. HUMAN FACTORS MODELLING & SIMULATION.....	60
4.1. SIMULATION DESIGN.....	60
4.2. HUMAN-FACTORS MODELLING	65
4.2.1. <i>Human Data Profiling</i>	68
5. AGENT DESIGN FOR EFFECTIVE REINFORCEMENT LEARNING.....	71
5.1. AGENT INTERACTION	71
5.2. DEFINING THE ACTION AND OBSERVATION SPACES	74
5.3. DEFINING REWARD STRUCTURES	77
6. AGENT PERFORMANCE EVALUATION	81
6.1. NEURAL NETWORK HYPERPARAMETER SELECTION.....	81
6.2. EVALUATION METRICS AND BENCHMARK.....	83
6.3. EVALUATION SCENARIOS	85
6.3.1. <i>On-Task Validation</i>	87
7. CASE STUDY	89
7.1. MANUFACTURING PROCESS MODELLING	90
7.2. HUMAN PERFORMANCE PROFILING.....	94
7.3. AGENT DEVELOPMENT AND TRAINING.....	97

7.3.1.	<i>Human Task Performance Prediction</i>	98
7.3.2.	<i>Reinforcement Learning</i>	100
8.	RESULTS	104
8.1.	TASK PERFORMANCE PREDICTION	104
8.2.	REINFORCEMENT LEARNING FOR COGNITIVE CONTROL	107
8.2.1.	<i>Reward Structures</i>	109
8.2.2.	<i>Parameter Variation</i>	110
8.3.	PRODUCTION PROCESS EVALUATION	112
8.3.1.	<i>Multiple Agents & Collaborators</i>	112
8.3.2.	<i>Human Data and Performance Prediction</i>	117
9.	DISCUSSION	121
9.1.	INTELLIGENT AGENT DEVELOPMENT	121
9.2.	INTELLIGENT AGENT PERFORMANCE	127
9.3.	HUMAN-ROBOT-INTERACTION	133
9.3.1.	<i>Ethical Use of Human Data</i>	138
10.	CONTRIBUTIONS, CONCLUSIONS & REFLECTIONS	147
10.1.	CONTRIBUTIONS.....	147
10.2.	CONCLUSIONS.....	149
10.3.	REFLECTIONS.....	151
	REFERENCES	153
	APPENDICES	163

List of Figures

Figure.1 - Illustration of the 5C's architecture used to quantify the capabilities and intelligence of CPS's (Reproduced from: (Lee, Bagheri and Kao, 2015)).....	21
Figure.2 - Agent structure in relation to the principle of rational belief and the principle of rational action. Reproduced from Baker & Tenenbaum (2014).....	40
Figure.3 - Existing information processing systems illustrated in terms of information flow through the system.	54
Figure 4 - The proposed framework illustrated in terms of information flow through the system. Divided into three functional layers.....	55
Figure 5 - Architecture illustrating information flow through the Agent, and how it interacts with the simulation environment.	57
Figure 6 - The model of interaction between human and robotic operators typically observed in a manufacturing process, each cell has an associated Cycle Time.....	61
Figure 7 - The model developed and the corresponding Anylogic simulation, each cell contains a delay and data capture function.	63
Figure 8 - Generic behavioural Statechart format to discretise robotic processes within manufacturing simulation. Multiple similar states may be defined, and each state corresponds to a chosen action.	73
Figure 9 - Visualisations of the configurations of different operators to be explored to ensure the robustness of the approach.	87
Figure 10 - Diagram of semi-automated production line used for case study. Call-out detail illustrates manual operation cell.....	89
Figure 11 - Case Study production line, illustrating the interaction between the robotic cell and human-operated cell, and the simulation model for the robotic operator developed in Anylogic.	91
Figure 12 - Statechart used within Anylogic to determine robotic operator behaviour, each yellow state is an action.....	92
Figure 13 - Heatmap illustrating the distribution of cycle time, mean, and standard deviation for all operators, across all conditions.	95
Figure 14 - Histogram showing the distribution of cycle times for all operators across all conditions.	96
Figure 15 - Derived parameters for each of the variation vectors considered. Means and Std. Deviations used to parameterise a gaussian distribution.	97
Figure 16 - Heatmap of RMSE scores, for the number of hidden nodes against the number of epochs trained. For a) 1 hidden layer and b) two hidden layers.....	99
Figure 17 - Hyperparameter grid search results for DQN Agent using a)10k training instances, and b) 25k training instances. RMSE scores represented by the colour (blue lowest).	100
Figure 18 - Illustration of the downstream automated process in the case-study production line, and the corresponding simulation model.	102
Figure 19 - Hyperparameter grid search results for downstream DQN Agent using a)10k training instances. RMSE scores represented by the colour (blue lowest).	103

Figure 20 - RO CT and HO CT over time for a) Monday, b) Tuesday, c) Wednesday, d) Thursday, e) Friday.....	105
Figure 21 - RO CT and HO CT over time for Monday shift of three weeks and corresponding shift orders a) 123, b) 231, and c) 312.....	106
Figure 22 - Results of DQN Agent training over 500 iterations for both the 5 Hidden Node and 10 Hidden node configurations.....	107
Figure 23 - Comparison of DQN Agents to benchmark case in terms of Idle Time, DQN score and Product Count.....	108
Figure 24 - Comparison of DQN structures i)5 Hidden Nodes ii) 10 Hidden Nodes, and training in terms of DQN score for different gamma values and reward structures: a) Penalty rewards, and b) Scaled rewards.....	109
Figure 25 - Breakdown of DQN performance over a 3-week simulation run incorporating different shift orders.....	111
Figure 26 - Comparison of DQN to benchmark (static case) for increased production demands.....	111
Figure 27 - a) Training over 500 iterations for the downstream operator, and b) Agent performance over the 3-week simulation run compared to static-case performance.....	113
Figure 28 - a) Training over 500 iterations for both the upstream and downstream operators, and b) Agent performance over the 3-week simulation run compared to static-case performance.....	115
Figure 29 - Training over 500 iterations for the agent with multiple human collaborators...	116
Figure 30 - a) Training over 500 iterations for the agent with profiled human task-performance, and b) evaluation compared to static-case.....	119
Figure 31 - a) DQN Scores over 500 iterations for the decision-making network, and b) Predictions vs observed human cycle times for analytical network for Monday of each week.....	120

List of Tables

Table 1 - Breakdown of the values used to modify the performance of each operator.....	67
Table 2 - Theoretical values measured during observation to enable profiling of performance for multiple human operators, over different conditions. Values assume a Normal Distribution.....	70
Table 3 - Values used to profile performance for each of the three operators based on observed case-study	94
Table 4 - Values used to parameterise the performance distribution for each of the three operators based on observed case-study.....	97

Acknowledgements

The completion of this work would not have been possible without the continuous support of a number of people. They know who they are, and to each of them I offer my most heartfelt gratitude. Thank you all for believing in me, especially at the times when I didn't.

Additionally, I would like to thank Olympus Surgical Technologies for the opportunity, and their support over the past few years, both financial and personal.

Abstract

This work is aimed at the understanding and application of several emerging technologies as they relate to improving the interactions which occur between robotic operators and their human colleagues across a range of manufacturing processes. These interactions are problematic, as variation in performance of human beings remains one of the largest sources of disturbances within such systems, with potentially significant implications for productivity if it continues unmitigated. The problem remains for the most part unaddressed, despite these interactions becoming increasingly prevalent as the rate of adoption of automation technologies increases.

By reconciling multiple areas encompassed by the wider domain of intelligent manufacturing, the presented work identifies a methodology and a set of software tools which leverage the strengths of neural-network-based reinforcement learning to develop *intelligent* software agents capable of adaptable behaviour in response to observed environmental changes. The methodology further focuses on developing representative simulation models for these interactions following a pattern of generalisation, to effectively represent both human and robotic elements, and facilitate implementation. By learning through their interaction with the simulated manufacturing environment, these agents can determine an appropriate policy, by which to autonomously adjust their operating parameters, as a response to changes in their human colleagues. This adaptability is demonstrated to enable the intelligent agents to determine an action policy which results in less observed idle time, along with improved leanness and overall productivity, over multiple scenarios.

The findings of the work suggest that software agents that make use of a reinforcement-based learning approach are well suited to the task of enabling robotic adaptability in such a way, and the developed methodology provides a platform for further development and exploration, along with numerous insights into the effective development of these agents.

1. Introduction

1.1. Background

While the word ‘robot’ conjures images of 1950s sci-fi, it originates in Czech writer Karel Čapek’s 1920 play, *Rossumovi Univerzální Roboti* (Rossum’s Universal Robots) as a term for ‘forced labour’. Perhaps ironically, it is as labourers, that robots have found their niche, and the increase in the prevalence of robotics and automation within the manufacturing sector in the wake of what has come to be referred to as the *third industrial revolution* has continued to the point where they have become ubiquitous (Moniz and Krings, 2016). Despite this, there remain a number of manufacturing tasks which require either a level of dexterity or adaptability that only humans are able to achieve. The result of this is a transitional period, in which a growing number of manufacturing processes feature tasks where robotic operators must interact with human operators as colleagues and collaborators (Teiwes *et al.*, 2016).

The investment in and adoption of automation technologies stems from the socio-economic pressures to continually increase the productivity of many European nations (manufacturing accounts for 25% of the German GDP and represents a considerable portion of the GDP of several other European Countries (Heinisch and Scheufele, 2017)). This is due to the need to remain competitive in the face of challenges including increased industrialisation and growth in the manufacturing sectors in the east (Goldstone, 2015); and an ageing population in many of these same European countries with developed economies. Investment has also been required to meet the massively increased demand within these developed nations for high quality, bespoke products (Tao *et al.*, 2011), developed using sustainable and efficient methodologies (Rusinko, 2007). As a consequence, the key to competitiveness in the modern age is resilience to, and capability to adapt in response to change. Therefore, existing approaches to manufacturing are rapidly becoming obsolete due

to their inability to efficiently provide the current requirements imposed by manufacturing systems (Leitão, 2009).

Concurrent to the developments in automation, and driven by the massive increase in data generation over the past several decades, advances in the field of computer science have resulted in modern software implementations capable of learning from vast reserves of data in addition to their own experiences to intelligently analyse new data to enable predictions, and decision-making (Witten *et al.*, 2016; Marsland, 2015). The application of these systems within manufacturing provides the capacity to overcome the challenges facing developed manufacturing industries, and exploration into the implementation of intelligent systems for high-level data analytics, data mining and machine learning within industrial manufacturing processes has already begun to be undertaken by many companies with large manufacturing requirements (Auschwitzky, Hammer and Rajagopaul, 2014).

This application of *intelligence* has given rise to various related fields, referred to collectively as Intelligent Manufacturing (Zhou, Liu and Zhou, 2015). Work in these areas often relates to the development of Cyber-Physical Systems (CPS's), which, as described by (Lee, 2008b), combine computational and physical processes such that the embedded computers can autonomously control, monitor and predict the physical processes on an automated level. The realisation of CPS's promise to deliver several benefits to manufacturing processes, through the combination of advanced digitalisation, simulation, and data analysis techniques, which enable manufacturing control systems to operate with greater fidelity and accuracy, and for advanced capabilities to be realised in the *intelligent* behaviour of autonomous systems, in terms of adaptability (Monostori *et al.*, 2016). These novel behaviours play an important role in facilitating the interaction between autonomous robotic operators and their human counterparts and have significant implications for manufacturing processes as a result.

1.2. Main Challenges

The role of the human operator is far from obsolete, particularly within heavily governed industries, such as logistics, food, pharmaceutical/chemical, and medical, due to stringent quality and accountability controls. Human beings are, however, subject to the influence of many factors which can dynamically and unpredictably affect their task performance, and its consistency, and consequently, they are a source of disturbance in such systems. The disturbances that arise as a result of the natural variation between human beings remain a significant obstacle to improving the productivity of these processes, and human components of such systems are often ineffectively modelled, leading to inefficiencies in the design and control of such processes. The study of these *Human Factors* is a field in its own right and is more comprehensively explored in section 2.5.

The impact of human performance variation is particularly pronounced in the case of manufacturing processes, where individual elements are highly dependent on one another, and ideal operation is typically realised as a perfect one-piece-flow of product through such systems, based on lean principles (Herron and Braiden, 2007; Adams *et al.*, 2001). Variation in human-beings, combined with ineffective modelling in the design of these processes, can often lead to a disparity between human task performance, and the repeatable performance of their robotic counterparts, with a negative effect on the overall system productivity. The factors which influence human performance are well understood from the perspective of human resources and management, However, limited work exists on how to how to apply this knowledge to the study of Human-Robot-Interaction (HRI), or how it may be leveraged to alleviate the effects of these interactions on manufacturing processes.

Within the manufacturing domain, work has demonstrated that the application of learning and intelligent data analysis to robotic operators within manufacturing processes may leverage novel capabilities to enable adaptable behaviour in robotic operators, as in the

case of the aforementioned CPS's. The ever-increasing application of intelligence to these systems has in itself developed into a new field, which studies how *interactions* between these robotic entities and their collaborating agents occur, known as *collaborative robotics* (Khalid *et al.*, 2016; Brown and Woods, 2017). Work in the field has resulted in insight and implementations of robotic operators that are able to coordinate actions and adapt behaviours to the state and actions of other operators, to achieve automation of more complex tasks, and further optimise existing ones in real-time, and in cases where modelling is difficult as a result of unpredictability. However, much of the research in this field has however neglected the importance of *Human-Robot-Interaction*, and its relevance to collaboration, particularly in the manufacturing setting. Consequently, it seems logical to extend this adaptability to the problem of mitigating the effects of the variation in the performance of human beings, and the resultant disparity.

This increased adaptability of robotic operators within a CPS enables changes in behaviour in response to the actions of others, potentially facilitating collaboration with human colleagues. Consideration of the impact caused by human factors can be used to model the consequent variations in performance, and a machine learning approach can be leveraged to enable the robotic agents to intelligently analyse the observed data, model the relationships between observed human performances, adapt its behaviour, and eliminate the variation in performance between operations, enabling processes to achieve a more optimal and agile one-piece-flow, in line with the modern application of lean-manufacturing methods (Monostori, 2014; Lee, Bagheri and Kao, 2015).

The theoretical aspect of this application is backed by this existing work, however, there are many substantial challenges involved in realising such a system. In addition to the existing questions surrounding the effective modelling of human beings within these scenarios, the technical implementation of the necessary data processing systems is

undefined, in both a practical and theoretical sense. Extension of machine learning techniques to enable adaptable behaviour in this context is currently poorly explored, as there is little knowledge on how learning models may account for human behaviour, evidenced by a lack of referenceable work in the area. Furthermore, there is a substantial challenge in developing an effective control system based on this modelling and analysis to effectively provide adaptability in these systems, as almost all of the elements and constraints of such a system remain unknown. What is clear, is that if the challenges associated with the development and implementation of such systems are met, there are potentially significant benefits to be realised in manufacturing processes, at both the human and system levels.

1.3. Identified Research Questions

Many of the emerging techniques that the field of intelligent manufacturing promise to enable the capabilities required to achieve this level of understanding and adaptability to be realised, from the digitisation of processes for modelling to the integration of full Cyber-Physical-Systems. Realising CPS applications has the potential to improve the collaboration between robotic operators and their human equivalents, across a variety of domains and applications, and with tangible benefits to manufacturing systems. Work in the domain of computer science, namely machine learning, has demonstrated that such systems may be realised through algorithms which are capable of learning from experience to determine individual behavioural policies to enable adaptable behaviours. Realising such a system requires a combination of methods, techniques, and approaches, but overcoming the associated challenges suggests such adaptability and the resultant benefits, may be attainable. From this, the following research questions have been proposed:

Question 1: How can the typical interactions between human operators and their robotic counterparts within manufacturing be modelled; and what implications does this have for the productivity of these processes?

This question is posed to develop understanding of how humans and robots work within manufacturing operations, as they occur in production line processes. As discussed, certain tasks remain that require levels of dexterity in control, and environmental analysis, that prevent their effective automation. Within production line processes, efficient systems are often determined by their *leanness*, and how close they operate to a one-piece-flow. The disparity in performance between individuals and across contexts is an unpredictable factor in the design, monitoring and analysis of these systems, and leads, often to sub-optimal process performance.

Question 2: Which methods can enable *intelligence* be implemented into the robotic operators in such a manufacturing system to enhance their understanding and adaptability with respect to the behaviours of their human counterparts?

Intelligent manufacturing is one of the fundamental areas under the Industry 4.0 initiative. Whilst not exclusively so, many of the key ideas are realised under the Cyber-Physical-Systems methodology, which combines physical process with computational elements to enable autonomous adaptable behaviours. Machine learning seems well placed to enable these capabilities, with reinforcement techniques having been demonstrated to allow intelligent agents to act in an autonomous and adaptable manner (Monostori *et al.*, 2016). To realise these systems, there remains a large volume of work to be completed, in terms of how to apply computational techniques, and model real-world processes, to provide the necessary

capability to extract knowledge for observations of the environment and others. This question aims to identify the appropriate use of intelligence to fully realise the potential of such systems, concerning improving the interactions between robotic elements with production line systems and improving the processes as a whole.

Question 3: Is it possible to leverage the benefits of an intelligent-agent-based control system for the robotic operators within manufacturing systems, to improve their adaptability and to realise benefits in interacting with human beings, and at the system level?

This question aims to explore novel concepts under the intelligent manufacturing umbrella, which seek to embed a degree of contextual and social awareness in robotic systems (social, in the context of this work, referring to the interactions and collective action of different individuals), to improve their interactions with humans, and provide improvements to these production processes. Enabling adaptive behaviours in robotic elements of manufacturing process has been hypothesised and demonstrated to provide tangible benefits in terms of optimisation (Lee, Bagheri and Kao, 2015; Scholze and Barata, 2016), but there is an identified lack of work on extending this adaptability, to enable robotic operators to adapt their behaviour based on the variation of human operators to reduce the impact of this disturbance on the overall process.

The research aims, that seek to be resolved through these questions, are well identified. Existing control systems for manufacturing processes are limited by the complexity required to enable the numerous benefits of mass data-collection; advanced computational intelligence; and advanced automation. Digitalisation of processes through data collection, decentralisation of control through both hardware and software, and the use of intelligent data

analysis, enable the development of control systems that are autonomous, adaptive and capable of autonomous response to change (Wang, Törngren and Onori, 2015). These capabilities are crucial to facilitating collaboration between elements of these processes which has been demonstrated to allow real-time optimisation of these processes, through adaptable behaviours.

Despite the promise of these systems to provide real-time optimisation and reduce the impact of disturbances on these processes, there has been limited consideration the human impact on manufacturing processes; an inherently unpredictable variable and a source of disturbance. As such, control systems must be developed, which have a contextual understanding of the factors that influence human performance, and which can act and adapt in a manner suited to working alongside their human counterparts, if the industry 4.0 vision is to be fully realised.

1.4. Thesis Structure

The following outlines the structure of this thesis, and the necessary elements considered to answer the identified questions. This opening chapter forms an introduction to the problem and wider fields and provides detail on the background and context which underpins this work. This is followed by a comprehensive review of the relevant literature covering five relevant topics: Intelligent Manufacturing & Industry 4.0, covering the wider field of computational intelligence and its application to the manufacturing sector; Decentralisation of Control, a topic within intelligent manufacturing concerned with improving the control of complex systems; Machine Learning, covering the computational implementations to extract knowledge from data; Human-Robot-Interaction, which explores the current state-of-the-art and theoretical background of this specialised area; and finally, Human Factors, focused on the theoretical background of human management and behavioural analysis.

The literature review is followed by a chapter which outlines the framework for approaching this research problem, and the relevant methods and techniques utilised to do so. This is followed by three chapters which constitute the main body of this research, which cover the modelling of human beings and the simulated environment, the design and development of the reinforcement learning agent, and the evaluation and validation processes of the relevant constituent elements, and performance at the system level. As a part of the identified process of validation, a case-study was completed, and the details of this are further outlined in the chapter immediately following.

The results of the previously outlined experiments are then presented to form a cohesive picture of the completed research and its consequences. These results are discussed and interpreted fully in the chapter following, which is sub-divided relevant to the identified research questions. These chapters are then followed by a final section of the work, which outlines the key conclusions, research contributions, and personal reflections on the project and its results. References and other information are then appended.

2. Literature Review

This chapter presents a review of all existing literature in this area and is divided into five main sections. These cover the scope of intelligent manufacturing, and its relationship to the current state of the industry; the decentralisation of control, and how this is a key enabler of collaborative systems; machine learning, and its application to enabling the *intelligent* aspects of such systems; the current state of the art of HRI, again with a specific focus on the relevance to manufacturing applications; and a final section covering human factors, and how this relates to modelling and prediction of human behaviours and variation.

2.1. Intelligent Manufacturing & Industry 4.0

Originally conceived as a strategy for economic growth in the manufacturing sector by a think-tank based in Germany, which consisted of engineers, business owners, politicians, and academics. The Industry 4.0 Initiative has, since its introduction at the 2013 Hamburg world fair (Zuehlke, 2010), emerged as one of the most widely discussed topics in the manufacturing world. The term, coined by the German government to describe their ongoing vision for manufacturing; illustrates the concept of ‘smart-factories’; manufacturing facilities within which each of the processes is connected either centrally or modularly to every other process and which makes use of computation and machine learning processes to autonomously self-adjust and self-correct to problems.

Whilst Industry 4.0 was the first and most globally recognised, similar initiatives have existed (the product of similar research founded on similar concerns) for years under the direction of the American, UK and Japanese Governments. Despite arguments that advances in the field of intelligent manufacturing, are simply a natural progression of technology, the initiatives have nonetheless served as a catalyst, coordinating research efforts and accelerating progress in recent years (Schönsleben, Fantana and Duchi, 2017).

The term *industry 4.0* has evolved beyond its original meaning, becoming, in many aspects, simply a buzzword that refers to the underlying aims and objectives –In the author's opinion- this devalues the term from an academic perspective, the 'simplification' that occurs with frequent (and often incorrect) use provides a useful terminology by which to group related research, and enables communication with the larger community.

Many of the advancements within Industry 4.0 have been driven by the emergence of 'Big-Data' (Hilbert, 2016). Developments in the field of big data have led to an understanding of how to handle and effectively process large datasets where statistical methods are no longer effective due to its complexity and volume. Despite advances in the field, the capture, handling, storage, and processing of large datasets remains challenging (Wuest, Irgens and Thoben, 2014; Babcock *et al.*, 2002). An extensive view on the challenges of practically managing this data can be found in (Jagadish *et al.*, 2014).

The capacity and understanding of how to process and extract of knowledge from this collected data has fuelled the emergence of a vast number of enabling technologies and fields of study which form a larger body of research (Brettel *et al.*, 2014), which is a cornerstone of the Industry 4.0 initiative. Termed Intelligent Manufacturing, the collective focus is used to describe the application and integration through a variety of disciplines to contemporary manufacturing systems (Zhong *et al.*, 2017; Li *et al.*, 2017; ElMaraghy and ElMaraghy, 2016). The main research focuses include: novel automation control systems, with) a focus on, decentralisation, virtualisation, reconfiguration, and adaptability (da Silva *et al.*, 2015; Mendes *et al.*, 2009; Shafiq *et al.*, 2016), the development and application of machine learning and artificial intelligences (Spezzano and Vinci, 2015); and virtual and augmented reality systems, which are being used to bridge gaps in geography, knowledge and skill level (Pirvu, Zamfirescu and Gorecky, 2016).

Whilst the area of Intelligent Manufacturing is itself a many-faceted problem, the recurring element that underpins much of this revolution is the collection, utilisation and understanding of data, or the study of ‘Informatics’, and all of the areas linked with the intelligent manufacturing research area rely on the capture and analysis of data in some capacity. To this end, the use of advanced data analytics and machine learning and the associated algorithm development and Software Engineering (Vyatkin, 2013), alongside embedded and distributed sensing technologies as in the case of Internet of Thing (IoT) technologies (Atzori, Iera and Morabito, 2010) have all seen a great deal of interest in recent years.

These areas build on ideas of big-data and connectivity, to enable digitalisation of manufacturing processes, where digital representations of real-world processes may be used for analysis, prediction, and control (Kritzinger *et al.*, 2018; Ding *et al.*, 2019).

The digitalisation of manufacturing processes is a key enabler of *Cyber-Physical Systems* (CPS), which form a combination both physical and digital elements, and a system of networked machines and sensors, which communicate and collaborate intending to enable autonomous intelligent behaviour (Lee, 2008a; Lee, Bagheri and Kao, 2015). The utilisation of data allows for these systems to construct detailed process models, and interpret perceived data with respect to these models, enabling:

Self-Awareness & Prediction; through the construction of a digital representation or *Digital Twin* of a system enables potential solutions and control structures to be virtually implemented and validated; and for improving understanding (Schuh *et al.*, 2014).

Self-Diagnosis; through data-driven modelling and intelligent analysis of real-time observations, which enables preventative maintenance and fault diagnosis, and reduces unpredictability and disturbances within the system (Chukwuekwe *et al.*, 2016).

Self-configuration; through the complex and distributed nature of CPS's, which enables self-configuration, as system controls, parameters, resources, and operations can be configured and reconfigured remotely, and in response to observed changes in the environment (Gurgen *et al.*, 2013).

Self-Optimisation; through adaptability and autonomy, which provide the tools necessary to implement self-optimisation. Experience-based learning can ensure process stability given changing contexts. This can be combined with digital representations to enable learning systems to explore states other than those observed to find ever more optimal solutions (Permin *et al.*, 2016).

The use of digitalisation by Cyber-Physical-Systems may be leveraged to improve manufacturing processes and products in various ways and provide them with many of the capabilities required under the Industry 4.0 initiative (Bagheri *et al.*, 2015).

The combination of these capabilities enables the *intelligence* of these systems to be quantified. The 5C's architecture ((Lee, Bagheri and Kao, 2015) outlines 5 levels of *intelligence*, and their associated technologies and capabilities. At the low end, the *Connection* level describes the majority of current manufacturing environments, that have adopted the principles of data collection, via connectivity, but whose systems are not advanced enough to provide reliable analytics. The levels of autonomy increase with the degree of intelligence -implying that these two factors have an inherent link with one another- with systems that achieve the *Configuration* level capable of adaptability and reconfiguration based on the perceived surroundings. This architecture is illustrated in Figure.1.

Current *CPS* implementations based on intelligence and distributed control have demonstrated adaptive scheduling, real-time modelling of processes, and Decision Support Systems. These features have been used to refine processes and component design across a

wide range of control disciplines, such as automation control for manufacturing, Electrical Grid control (Pipattanasomporn, Feroze and Rahman, 2009); and Traffic Management/Air Traffic Control systems (Wang, 2005).

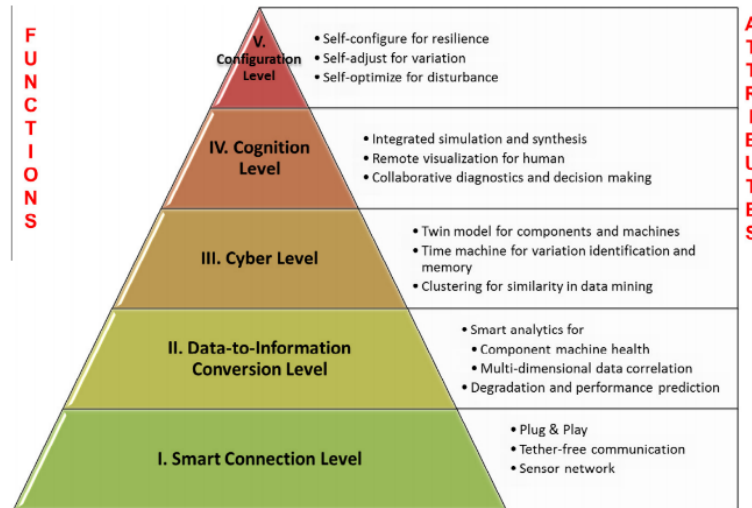


Figure.1 - Illustration of the 5C's architecture used to quantify the capabilities and intelligence of CPS's (Reproduced from: (Lee, Bagheri and Kao, 2015)).

The development of the necessary technologies has enabled easy experimentation with these systems; however, as a result of a wide variance in their application, large scale implementation and commercialisation is not without challenges. Standardisation is essential for the implementation and facilitation of interconnectivity; a key component in enabling CPS. Standardisation of electro-mechanical components, software, interfaces, communication and interoperability protocols, data formats, and system intelligence are all critical (Leitão and Strasser, 2016; Weyer *et al.*, 2015). The other major challenge is security, as Cyber-Physical Systems have multiple vulnerabilities both to external security threats; which can exploit communication networks, and information security protocols, which can lead to internal system failures and error. Consequently, both be addressed to ensure reliability, protection and resilience against these threats (Pasqualetti, Dörfler and Bullo, 2013; Ali *et al.*, 2018).

In addition to overcoming these challenges, several other areas remain in which more work is needed to fully realise the potential of these systems (Wang, Törnngren and Onori, 2015). Given the potentially unlimited number and variety of applications for CPS's, robust, developed architectures and methodologies for their implementation are lacking, and development of the necessary guidelines and standards to facilitate implementation and development of CPS's is also a key challenge in their effective realisation. The increase in both the number of computational elements and process elements also inherently provides a major obstacle to realising Cyber-Physical-Systems, which rely on distributing the control of individual elements to overcome increasing system complexity. Work on coordinating these networked elements or agents is crucial for enabling flexibility and adaptability through self-organisation and context-aware decision-making. The use of high-level analytics will enable robust control, capable of dealing with errors and unfamiliarity, and adaptability, through an awareness of contextual factors inherent in the decision-making process. Intelligent agents have been utilised in practical applications and demonstrated as capable of performing a multitude of self-organisational and autonomous tasks for use in the manufacturing industry (Zhang *et al.*, 2016; Musil *et al.*, 2017).

2.2. Decentralised Control

Many of the existing manufacturing control systems were developed at the beginning of the digital age, with the emergence of robotic and computational manufacturing systems. These systems are frequently centralised and use a hierarchical control structure, which, due to its intrinsic capacity for optimisation, led to significant increases in productivity over traditional methods that were in use at the time. However, as the representations of production processes become increasingly detailed and complex, top-down control faces significant challenges when faced with enabling autonomous adaptable behaviour (Cao *et al.*, 2013; Leitão and

Restivo, 2006). Traditional automation techniques have been made obsolete by the rise of the data-driven processes, and the consequently increased complexity required of *intelligent* systems.

To overcome these challenges, manufacturing control systems must become distributed in terms of both hardware and software; and intelligent, through the application of machine learning and artificial intelligence techniques. This will enable said systems to respond dynamically to change in response to observations of the environment, flexibility, reconfigurability, adaptability, and autonomy, to increase productivity in line with the aims of intelligent manufacturing (Leitão, Mařík and Vrba, 2013; Brettel *et al.*, 2016).

The concept of distributed control and the mechanics of distributed interaction have been studied and utilised since the introduction of *planning agents* in the 1970s (Wooldridge and Jennings, 1995). Contemporary applications explore intelligent agents, often form combinations of hardware and software (these combined elements form a logical unit within the system, defined as a *Holon*, within the Holonic Manufacturing System) (Van Brussel *et al.*, 1998). Each agent is autonomous with its own sensory inputs, objectives, knowledge, and internal intelligent functions. The strength of distributed control when faced with complexity, is that it enables a complex problem to be divided into several small problems, each of which is distributed to a network of multiple intelligent agents. These agents interact as governed by the rules of the Multi-Agent-System (MAS). And the global control decisions are determined by the collaborative efforts of the agents through their interaction, as no single agent has a global view of the system (Leitão and Colombo, 2006). Each of the agents solves their own localised problem, aiming to facilitate those around them, to provide a global solution to the original global problem.

In the manufacturing context, a multi-agent-system typically consists of agents of two types: *Resource*, represent hardware, such as machines and robotic operators; and *Orders*,

which represent raw materials, tooling, components, etc. Software agents, which are able to act autonomously based on observation of the environment and communication with other agents, are connected to the physical hardware of the robot forming a *Resource Holon*, which enables actions to be taken based on computational decisions. These resource holons are similar to the elements defined in Cyber-Physical Systems, which also form combinations of hardware and software. The formation of these holonic systems enables physical hardware to be controlled in a distributed and adaptive manner, as the software interacts and generates commands as necessary to achieve a given state or goal. For a detailed breakdown of the history, structure and capability of Intelligent agents and their applications see (Marik and McFarlane, 2005).

The challenge of coordinating multiple agents effectively and implementing control systems based on these methods, has resulted in the emergence of *collaborative robotics* as a distinct discipline. Collaborative robotics is based on improving the interaction dynamics of robots with others, by leveraging the capabilities of the intelligent agents to overcome the former roadblocks to centralised approaches. This is, therefore, of particular relevance to the development of distributed manufacturing control systems, which are heavily reliant on robotic systems, and how they interact with one another, and human operators (Bochmann *et al.*, 2017; Kirgis, Katsos and Kohlmaier, 2016). Research in this area primarily focuses on how to enable intelligent processing of sensory data in terms of coordinating robotic behaviour with that of others. This requires consideration of the intelligent agent's control structure, and how different functionality may be used to enable an adaptable behaviour and appropriate response to changes in the environment, or the actions of the collaborators (Vorotnikov *et al.*, 2018; Khalid *et al.*, 2016; Brown and Woods, 2017).

The coordination with other agents in the system inherently requires the consideration of the mechanisms of collaborative behaviours and the internal structure of learning and data

analysis. Whilst of course teamwork is not unique to Homo Sapiens, collaboration is commonly considered a very human trait, many studies on collaboration have sought to understand cognitive processes as they occur in humans, and to replicate these *cognitive* processes. The combination of these cognitive processes to enable an agent to behave intelligently has led to the development of numerous *cognitive architectures* (Kotseruba, Gonzalez and Tsotsos, 2016).

Many of these architectures exist to provide intelligent agents with *cognition* (Haykin *et al.*, 2012). Notable examples of these architectures include ACT (Anderson, 1996), SOAR (Laird, Newell and Rosenbloom, 1987), and C4 (Isla *et al.*, 2001) which was originally used to provide AI for enemies in video games. These architectures contain many insights with regards to the agent's structure and how it interacts, despite being conceptualised for more primitive computational systems and hardware. These insights may be drawn from both the variance and similarities in their designs, with a common feature among many being a modularised structure, with multiple elements which are responsible for different aspects of cognition and interact to form the desired behaviours. The modules contained within each architecture are often structured around a central module which acts as a controller for internal processes and acts as the decision-maker. This control module is supported by others which facilitate necessary behaviours; such as Perception, Learning, Decision-Making, and Memory. The segregation of processes in this way facilitates the integration of low-level perceptual and motor control with higher-level knowledge extraction and decision-making processes (Salvucci, Boer and Liu, 2001) through a further reduction in the complexity of control. Such an approach is therefore beneficial in terms of combining higher-level processing elements with established control techniques without interference.

The final concept with an application to distributed control that comes from the study of cognition is *embodiment*, a factor which is beneficial in terms of developing software

agents, and how they interact, for several reasons. Embodiment refers to the clear distinction between internal and external processes and factors. This embodiment enables multiple agents with identical internal structures, to behave based on their individual goals and knowledge gained from their own cumulative experience, and its context (Hoffman, 2012). This enables these agents to behave differently from one another, even when provided with the same observation, due to the unique nature of each agent's cumulative experience (Young *et al.*, 2011).

In a practical sense, the implementation of embodiment is well suited to the structure of *Object-Oriented* programming languages (this has benefits with working with languages such as C++ and Java, two of the most widespread programming languages, especially within the manufacturing industry), which defines pieces of code as individual objects, with defined internal and external access.

Despite a renewed interest in the application of distributed control, there remain many problems to be resolved to enable the use of these methodologies. These include questions surrounding practical issues, such as connectivity, and security (Ali *et al.*, 2018; Bannour, Souihi and Mellouk, 2018), but one of the key challenges facing implementations of intelligent agents is a lack of standardisation, both in a real sense, but also in terms of agent behaviour and design, due to the application-specific nature of many approaches. Some of the most respected of numerous architectures and frameworks include ADACOR; developed by (Leitão and Restivo, 2006), a holonic architecture which has gained traction and popularity due to its applicability to manufacturing systems; And GRACE; or inteGration of pRocess and quALity Control using multi-agEnt technology, a Multi-Agent-System developed with the purpose of utilising collaboration and distributed control for process and quality control integration (Leitao *et al.*, 2015; Rodrigues, Pereira and Leitão, 2013).

A manufacturing control system that satisfies the conditions and overcomes the challenges of decentralisation has numerous advantages over traditional centralised control systems. Multi-agent systems are a promising implementation of distributed control, through embodiment of learning algorithms to provide *intelligence*, will contribute to increasing capabilities and the realisation of CPS. Recent applications of *intelligence* to autonomous agents have made considerable use of the strengths of a neural network-based approach to automate each agent's analytical and decision-making processes, based on its cumulative experience (Monostori, 2014; Monostori *et al.*, 2016; Shi *et al.*, 2020).

2.3. Machine Learning

It is important to consider the mechanisms which enable the capacity for *intelligent* behaviour. Whilst *Intelligent Manufacturing* as a field of study is well defined in its scope and incorporated areas, *Intelligence* is a subjective term, and the definition used within the field is fluid. Typically, intelligence manifests itself through several behaviours: the ability to observe a set of inputs and use this information to take appropriate action. Attempts to quantify such behaviour goes back as far as the 1940s, and Alan Turing's now infamous 'Turing Test'; designed to establish whether a machine was capable of human-level intelligence, through its decision-making abilities (Turing, 1950). Developments of both hardware and software, have enabled machine learning techniques to become ever more powerful, and increasingly feasible to implement on accessible hardware (Abadi *et al.*, 2016; Team, 2018), and for Artificial Intelligences to be applied in a wide range of applications, including Image Recognition; Language Translation; Finance; Logistics; and Gaming (Silver *et al.*, 2016; Firoiu, Whitney and Tenenbaum, 2017; Witten *et al.*, 2016).

Machine learning as an approach is increasingly leveraged as an effective tool for the generation of knowledge from data (Lee, Bagheri and Jin, 2016; Miškuf and Zolotová, 2016),

and has been explored extensively within the manufacturing field (where reserves of historical data are plentiful) to generate process insight; and more recently, to realise the ideas presented by Intelligent Manufacturing (Sharp, Ak and Hedberg Jr, 2018; Qiu *et al.*, 2016).

Owing to their strengths in abstraction and pattern recognition; their ability to manage a large number of data inputs; and their adaptability to a wide variety of applications, neural networks have been successfully applied to learning and analytics tasks, within several fields and applications and provide a non-deterministic method of matching a number of input variables to singular or multilabel outputs, and for approximating relationships between multidimensional data. Neural Networks and their applications are well established and are documented in (Witten *et al.*, 2016; Goodfellow, Bengio and Courville, 2016; Schmidhuber, 2015).

A neural network is a collection of *nodes*, each of which is connected to numerous other nodes through weighted connections, analogous to human neurons. The structure and interactions of the network can be used to exhibit a multitude of desired behaviours. These Neurons are typically arranged into *layers*, forming representations, with the leftmost layer *Input Nodes*, which convey information into the network, and the rightmost layer the *Output Nodes*, where the resulting output is provided to the system. The observation signal is propagated through the network, based on the *activation function* of each neuron, which describes its behaviour in terms of whether it passes on the signal (Baldi and Hornik, 1989).

In a typical implementation of *Supervised Learning*, in which a set of *Training* data is used with known input/output pairings. The weights are initially randomised and are altered during training, usually via Gradient Descent backpropagation, which aims to maximise the rate of correct output for a given input and minimise the total system error (Medsker and Jain,

2001). Significant care must be taken when training Neural Networks, as gradient descent is particularly vulnerable to local minima phenomena (Baldi and Hornik, 1989).

The network may also contain a vast number of intermediary layers or *hidden layers*, which, as each forms a representation of the input based on the layers before it, may be designed to provide layers of abstraction to an input example. Neural networks are defined in terms of these layers, whereby increasing the number of layers, increases the *depth* of the network. The addition of these intermediary hidden layers proved to be monumental in the development and application of Neural Networks to advanced and *Intelligent* tasks.

Termed *Deep Learning*, the increase in intermediary layers enabled the development of advanced techniques which have proved to surpass most other methods in the discovery of intricate patterns in datasets large in volume and high in dimensionality (Goodfellow, Bengio and Courville, 2016). NN's utilise the multiple levels of layers to *abstract* data and form representations of different features in the data. *Convolutional Neural Networks*, which use hierarchal layers of tiled convolutional filters, to reduce high-dimensional data into increasingly abstract representations. These networks are capable of identifying policies with little prior knowledge and have been used successfully in the field of machine vision and language processing (Mnih *et al.*, 2015; Matsugu *et al.*, 2003). Additionally, Recurrent neural networks make use of deep architectures to incorporate elements of short term memory, and can accurately track temporal changes from one state to the next, and improved pattern recognition over time from time-series data; with demonstrated success in speech and written word comprehension and generation (Medsker and Jain, 2001; Sak, Senior and Beaufays, 2014).

An important strength of neural networks is their capacity for *Unsupervised Learning*, reducing the need for extensive examples required to train the network; and hand-designed feature extraction and dataset manipulation (LeCun, Bengio and Hinton, 2015). This enables

an iterative approach to training to be used, to enable a neural network to make associations between states and sets of actions, through the use of rewards and penalties provided whilst the agent is training and exploring an environment. Within control theory, this is referred to as Reinforcement learning, and as mentioned, is commonly utilised to dictate the behaviour of autonomous agents within multi-agent-systems, through control of agent actions and process parameters, in response to changes in the external environment as a result of action (Permin *et al.*, 2016). This learning is referred to as online if the learning is conducted directly in the problem setting, using real-time data, and is used to autonomously control to real-world agents; whereas offline learning makes use of simulation and is used for analytics, and when the training may require millions of instances, impractical to implement on an onboard controller; and allows control schemes to be evaluated in an isolated, and therefore safe environment (van Otterlo and Wiering, 2012).

Application of neural networks are vast, however recent implementations have demonstrated success at the combination of *Markov Decision Processes* (MDP's) (Bellman, 1957) and Neural Network learning for Multi-Agent-Control (Lowe *et al.*, 2017; Sutton and Barto, 2018) in developing agents trained via Reinforcement. The interaction of devices in the Cyber-Physical-System environment provides opportunities for decentralised control systems. The MDP model, separates the system into a state, s , representing the condition of the environment at different timesteps; and a set of actions, a , which the agent may perform, before receiving a reward, r , and an updated state observation, s' . Actions are selected based on a policy, π , which aims to maximise the cumulative expected discounted sum of rewards. In an MDP, the question becomes defining the policy to maximise the received reward. In Q-Learning, this is achieved by assigning each action a value function $Q(s, a)$, or 'Q-Score', representative of the quality of an action in a specific state (van Otterlo and Wiering, 2012; Watkins and Dayan, 1992). These scores are generated using the Bellman Equation.1:

$$Q(s, a) = \sum r_t + \gamma \cdot \max_a Q(s_{t+1}, a) \quad (1)$$

These scores may then be used to select the appropriate action for a given input. This approach has proven exceptionally competent at learning complex and non-deterministic systems, especially when leveraging Neural Networks to approximate the policy in an MDP, by predicting rewards for actions given a state based on cumulative experience of interacting with the environment; referred to as Deep Q-learning Network's (DQN's).

The success of the application of DQN's to learning tasks is as a result of the ability of neural networks to act as function approximators and to evaluate their performance based on received feedback, which minimises the need for labelled data, and the need to fully define the environment and its possible states (Hester *et al.*, 2018; Chen, Ying and Laird, 2016; LeCun, Bengio and Hinton, 2015). More recently, Google's Deepmind, have focused on developing networks that outperformed existing examples and human performance across the domain of multiple ATARI 2600 games, which has become a benchmark for learning systems to learn from visual input, with performance over multiple games being used as a good measure of a networks generalisability. This work is a natural extension of Google's existing work in the field of image recognition and was one of the first applications to utilising the strength of convolutional nets to extract abstract representations from an array of on-screen pixels and their values in different colour channels which are used to form digital images. This ability to create simple representations from high dimensional inputs further highlights the strength on neural networks to effectively learn from complex state information and has since been applied across varied domains, including manufacturing (Kusiak, 2019).

These tasks extend the application in terms of how neural network predictions may be used, by extracting state value functions from visual state representations. In addition, the work by Deepmind has led to the establishment of multiple techniques exist to improve DQN

performance during training. These include the use of training methodologies such as advanced action-selection policy methods, including epsilon decay, which uses an additional parameter, *epsilon*, to encourage initial exploration of the environment by selecting random actions, which decays over time to give the agent progressively more control; *Experience Replay*, where a buffer of recent transitions is collected and sampled for training, serving as a form of memory; and the use of Fixed Targets, which involves using a separate *target* network to make value predictions for the training of the action-selection network, which enables the algorithm to converge to a stationary target. The Target Network is then updated after a certain number of training steps (Mnih *et al.*, 2013).

The application of DQN's to a wide variety of applications is well documented in recent work, however, there remain challenges in the effective utilisation and training of the approach. As with other Neural Networks, DQN's are similarly susceptible to converging on a suboptimal learning policy, as a result of their relatively fast response to prediction errors which results in a susceptibility to local-minima phenomena (Bernstein and Zilberstein, 2014), which may prevent the network finding an optimal solution. There are, however, various novel techniques and methods for effective training in a reinforcement learning environment. These methods typically apply to DQN's and associated applications which can leverage offline learning in a simulated setting, enabling the network to perform multiple training iterations and converge to an optimal policy. The training of algorithms iteratively is a time-intensive process however, with many image-based training regimes requiring millions of iterations. Similarly, to defining neural network structures and training in a supervised fashion, there are several additional parameters which must be defined a-priori to the learning process. The digital nature (or increasing capacity for digitalisation of many applications) of the problems DQN's are tasked with solving does in itself enable several novel techniques to further leverage simulation and offline learning to overcome many of

these issues and provide more effective training processes. Notable examples include Actor-Critic Architectures (Sutton, 1985; Sutton and Barto, 1998), which compare the performance of multiple networks and propagate the most successful changes; and approaches using principles of Natural Selection as inspiration, the most well documented of these approaches being *Generative Adversarial Networks* (GAN's) (Goodfellow *et al.*, 2014). Other *evolutionary* algorithms mutate the network topology randomly during their training, with mutations that lead to a better performance being retained, and vice versa. This use of the evolutionary process for training neural networks, has itself generated a new 'spin-off' field of computer science and algorithm design, with similarities to many biomorphic design applications.

The popularity of Neural Networks, in addition to continual advances in hardware speed and power under Moore's Law, has resulted in the development of robust development platforms and architectures for experimenting with and implementing Neural Networks. Notable amongst these examples is Google's *Tensorflow*; a Python software library focused on the development of NN's, with API's for Java and C++. The platform has been used with increasing success in the field of neural network development and is capable of many specialised & large-scale intelligence tasks (Abadi *et al.*, 2016), M. et al. 2016). Numerous other software packages and implementations exist for a variety of languages and with a variety of different capabilities, DL4J is a robust, Java-Based package with an active support community, along with more general machine learning packages including WEKA (Witten *et al.*, 2016), and KERAS (Chollet, 2015).

The development of robust, open-source software libraries has greatly accelerated the rate of knowledge advancement in the area, although practical implementation is not without challenges. The increased algorithm complexity has consequences for computation; in terms

of both time and loading; and presents challenges in management and analysis of data (Jagadish *et al.*, 2014), in the interests of achieving real-time control

The strength of neural networks, especially in applications with complex and unpredictable or random factors affecting their dynamics is the iterative or evolutionary nature. The capacity to learn from previous experience and generalise this knowledge into new areas has proved a crucial ability in developing many advanced robotics techniques.

This new approach to learning, using an unsupervised and iterative approach has made real several *biomorphic traits*, which have important applications in robotics. The US robotics company ‘*Boston Dynamics*’ is one of the pioneering robotics companies in the world today; developing and consistently refining a range of advanced robots, which replicate the musculoskeletal structure of biological creatures to overcome issues associated with mobility, and crucially, have exploited the strengths of neural networks to provide control over coordinated perception and motion (gait and balance etc.).

This has enabled a number of successful robots to be developed, including: ‘Big-Dog’, a quadrupedal robot developed for the US military for traversing rough terrain, providing transport for combat equipment as part of a unit of troops (Raibert *et al.*, 2008). The increased mobility and all-terrain capability and the versatility that the platform provided by Big-Dog has more recently been miniaturised and modularised into SPOT (Boston Dynamics, 2020b), a commercially available fully autonomous quadruped, aimed at industry. Shipping in the first quarter of 2020, the company hopes that facilitating access to the design will itself generate new ideas and areas of study.

The most prominent and well known of *Boston Dynamics* robots is, however, ATLAS (Boston Dynamics, 2020a). ATLAS is anthropomorphic in structure, facilitating its application to human-based tasks, and is designed to provide a general task versatility by being constrained to a similar range of motion enhanced by increased strength and durability.

The development of anthropomorphic systems like this provides many benefits, ATLAS can lift and carry objects, manoeuvre over obstacles whilst doing so, and regain and correct its balance if interfered with. Applications include replacing many manual, menial tasks, and replacing human beings in situations too hostile for human beings to survive (Hsu and Peters, 2014); as evidenced by the ATLAS project's funding from the Defence Advanced Research Projects Agency (DARPA). Anthropomorphism does, however, present significant challenges in the form of the Uncanny Valley phenomenon (Mori, 1970; Young *et al.*, 2011), which is often a significant factor in project failure.

Tangent into the benefits of biomorphic applications in robots aside, neural networks themselves are an excellent demonstration of the strengths of biomorphism in design. Neural networks enable human-like traits and abilities by learning as we do (Fink, 2012). Indeed, the fascination with anthropomorphism in robotic design dates back centuries to the intricate clockwork *Automatons*, (designed and built to give the illusion of human behaviour), but as demonstrated, has important implications for contemporary robotics, and the way in which we interact with them. Robots that move, look, and think as we do are inherently easier to predict, understand, and ultimately trust.

The autonomy and adaptability that machine learning enables can be argued to increased *intelligence*, and the capacity to which it can be implemented within manufacturing systems. The benefits that intelligent behaviours enabled by Neural Networks in a reinforcement learning environment are proven by their frequent appearance in the literature on learning tasks, and create new opportunities, as behavioural models that rely on learning and experience to dictate their outcomes can be embodied by these techniques. Combined with the principles of distributed control, it enables many of the capabilities envisioned under *Intelligent Manufacturing* to be realised. It is, however, both appropriate and necessary to

consider these capabilities, with respect to the human element of the systems that remains, and how they may be leveraged to facilitate interactions between human and robotic entities.

Furthermore, the utilisation of neural networks as a learning model has significantly expanded the field within computer science which specialises in understanding and replicating human thought processes, known as cognitive computing. Intuitively, approaches within cognitive computing make use of combinations of neural networks, to replicate cognitive processes (Noor, 2015; Modha *et al.*, 2011), and to replicate human behaviours. This has potential implications for the facilitation of collaborative behaviour and the improvement of human-machine-interaction. With recent work on social cognition and social intelligence suggests that providing intelligent robots with social understanding, and human-like cognitive processes and structures, will better enable natural and intuitive behaviour when interacting with humans (Vernon, 2014; Wiltshire, Barber and Fiore, 2013). This is further explored in the following section.

2.4. Human-Robot-Interaction

The study of interaction mechanisms between humans and their robotic counterparts has seen rapid growth, as a consequence of the increased capabilities of automation, resulting in an increased prevalence of HRI's in the manufacturing setting. Furthermore, this interaction is of increasing complexity, as intelligences become more capable. As discussed in the previous section, enabling robots to behave in a way that is adaptable as a response to changes in human behaviour has considerable significance and application to improving these interactions. There is a growing interest in the field, but limited work into its applicability to robotics under the intelligent manufacturing paradigm, in contrast to other areas of investigation (Lemaignan *et al.*, 2017). This section discusses this work in detail and considers the current state of the application of intelligence to robotics where they interact

with humans, in terms of both current approaches and interesting research opportunities that should be exploited.

It is also important at the beginning of this section, to make clear the distinction in terminology that surrounds this area, as many different phrases and nomenclature exist within the literature. Broadly speaking, *Human-Machine-Interaction* (HMI) and *Human-Robot-Interaction* (HRI), are used to refer to interaction at the abstract level, covering all forms. However, the concept of interaction may be further divided into two types: **Direct Interaction**; which are types of interaction that involve direct physical, communication or manipulation. These types of interaction include: User-Interfaces (UI's), both Physical (PUI) and Graphic (GUI) User-Interfaces, and physical interactions involving robotics hardware, and **Passive Interaction**; which are types of interactions that involve passive communication. These types of interaction are concerned with more abstract behaviours associated with inferring knowledge and understanding based on both social and contextual cues.

There are numerous contemporary implementations which exist to demonstrate and leverage the benefits of intelligence (in terms of facilitating collaborative and assistive behaviour in robotic operators) in Human-Robot-Interactions and provide entirely new opportunities. Based mostly on advances in machine vision systems and improved image processing, current implementations focus heavily on *Direct Interaction*, and typical interactions include:

Multi-Robot Handling/Robot-Human coordination; where the end effector coordination of multiple robots can be used to increase human strength, enabling handling of large and unwieldy components, and reducing the manual aspect of human Labour (Djuric, Urbanic and Rickli, 2016); Coordination with the motion of a human operator additionally improves safety when sharing a common work area, and collision detection is implemented in commercially available robotic operators manufactured by KUKA (AG, 2020) and ABB

(ABB); enabling them to share a work environment with a human operator with increased safety.

Direct-Teaching; where the use of collaboration enables proximity working, and programming of advanced motion combines the flexibility and reconfigurability of humans with the strength, accuracy and repeatability of robots. Applications enable autonomous replication of advanced manufacturing processes, such as composite layup, oversee component handling, and welding fabrication (Eckardt, Buchheim and Gerngross, 2016; Agravante *et al.*, 2014; Rozo *et al.*, 2016; Wang and Zhang, 2017).

Augmented/Virtual Reality; used for visualisation based on virtual models, can be used for remote instruction, training, and real-time information retrieval/delivery; for example, technicians may be provided with instructions on procedures based on 3D CAD data, which can be used to provide fully virtual training environments and simulations, or be overlaid onto images of the real-world problem and used to instruct (Danielsson, 2016; Paelke, 2014).

Perception; leveraging improved image and data processing technology, including the use of machine learning, has enabled many modes of perception. Current research involves voice recognition and natural language processing (NLP); facial recognition; gesture control; and motion tracking (Mohammed, Schmidt and Wang, 2016; Matsas and Vosniakos, 2015).

However, it is the development and application of *intelligence* to robotic systems that have elevated the role of the machine from a subservient entity to one with its own goals and motives, and the ability to make independent decisions to achieve them. This has, in turn, led to the current state of research, in which robotic entities interact on the social level, forming cooperative and interdependent relationships with others (Warta *et al.*, 2016). As the learning ability of intelligent agents improves and their decision-making becomes more capable, the interactions and relationships that exist between humans and machines grow

more complex and based more on the inference of information from contextual clues; a behaviour more typical of *passive interaction*.

Therefore, to facilitate interactions between humans and robots at the social level (where they can adapt their behaviour to be more cooperative or beneficial to their collaborators) leads to an argument for robots to have the capacity to think in the way human beings do, as the development of intelligence which is based on the ability to observe, perceive, and communicate both leads inherently to social interaction, and is already similar to the models found in human learning. Consequently, by developing behavioural models based on our understanding of human cognitive processes, and how these relate to social behaviour, it is anticipated that robotic systems that are able to mimic or exhibit human-like behaviours; reactions; motivations; and tendencies will interact intuitively, through the display of appropriate concurrent behaviours (Vernon, 2014), facilitating these interactions with human beings. This has particular relevance in how agency is implemented in robotic systems when interacting with human beings. (Wooldridge and Jennings, 1995) provide two different concepts: The *Weak* and *Strong* notions of agency. The weak notion of agency describes any system that exhibits the following behaviours: *autonomy; social ability; communication language; reactivity; and pro-activeness*. The strong notion of agency describes a system capable of weak agency, but that is designed or implemented in such a way as to mimic anthropomorphic features or behaviours, to facilitate interaction.

The modelling of *social cognition* (or the cognitive processes which govern social behaviour and the process of decision making based on observation of others) requires a functional understanding of the mechanisms in the human context; fortunately, these have been well studied in the field of Psychology. Within the literature, the concept is frequently addressed by considering the social interactions of agents, an overlap occurring as a result of their aforementioned suitability for dynamic interactions. In terms of how human beings

perceive a situation and form an action based on it, models commonly embody perception through *the principle of rational belief*, the assumption that the agent will hold sensible beliefs, and that beliefs are formed based on these principles from observations of the environment; and reaction through *the principle of rational action*, which assumes that the

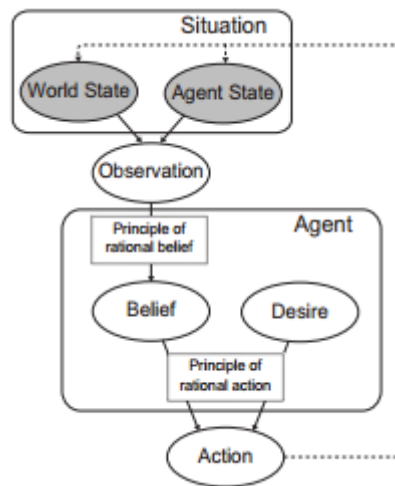


Figure.2 - Agent structure in relation to the principle of rational belief and the principle of rational action. Reproduced from Baker & Tenenbaum (2014).

agent will take appropriate action in relation to how it's observations match up to its beliefs and desires (Baker and Tenenbaum, 2014). This concept is illustrated in Figure.2.

The structure of this model bears a remarkable similarity to the application of a DQN and training under an MDP of an intelligent agent in the context of learning for distributed control, lending further weight to this approach. The agent receives an observation that describes the state, which forms a belief (Q-scores) when combined with the agent's principle of belief (Neural Network). This enables the agent to select an action based on the policy of rational action (or the decision policy, π , of the MDP). In addition to considering how cognitive processes may perceive and act based on inferred information from the environment to enable interaction, it is also useful to consider the internal processing of this information. From the perspective again of modelling human cognition opinion is split

between several main branches based on the distinction between two types of *cognitive processing*; *Type 1* and *Type 2* (Evans and Stanovich, 2013).

Type 1 processes are typically dominant where rapid response time is required. Influencing factors include a multitude of available clues, presented simultaneously; and a degree of familiarity with the situation and the agents involved, including knowledge of situational outcomes. Inversely, *type 2* processes occur in situations where response time is non-critical, and are typically more analytical in nature; focusing on specific relationships between a relatively small number of cues, that may be provided dynamically or in a specified sequence; The situations are typically unfamiliar to the agent and analytical reasoning is employed to identify patterns and relationships, to form appropriate response behaviours (Wiltshire *et al.*, 2016; Wiltshire *et al.*, 2014). Traditional automation control, specifically its responsive and reactionary nature is analogous to instinctive *Type 1* processing, whereas the recent application of intelligence to these systems has been achieved through learning and analysis, which are more typical characteristics of *Type 2* processing. It seems likely, therefore, that applications of intelligent manufacturing will retain some level of traditional automation control to govern *type 1* cognitive processes, combined with an additional cognitive process, based on analytics of the whole system, representative of *type 2* processes. The manner in these two types of cognitive processes interact to produce an accurate model of cognition is a source of debate, but there is significant confidence in Dual-Process Theory, which considers both types of cognition to contribute to exhibited behaviour, but to occur independently of one another, reducing computational load and reliance on large working memory.(Evans and Stanovich, 2013). As a result of their fast processing, default responses are governed by *type1* processes; unless superseded by higher-order reasoning though processes, or *type2* process. This work lends weight to the argument that traditional autonomous control should be retained, and control mechanisms developed to supersede the

instructions and control of these systems when necessary, based on higher-level, data-driven, decision-making.

As a note on quantifying the *soft benefits* that facilitating robotic behaviours based on observations of a collaborator, the following section describes the impact of human perceptions of robots. (Young *et al.*, 2011) observes that human beings inherently apply anthropomorphism to robotic entities, ascribing them personalities, and attributes such as names and gender. This tendency to socialise with robotic entities can be seen to be prevalent even when there are no specific design features to incite such a response (Forlizzi and DiSalvo, 2006). This is due to several factors, which set robotic technology apart from other technological artefacts; robots can work directly and in close proximity with human beings, within the bounds of personal space, but more importantly, they can do so autonomously, promoting the illusion of intentionality (Young *et al.*, 2011; Dennett and Haugeland, 1987). This intentionality helps people to build expectations and predictions of behaviour, with the result of increasing familiarity and a sense of *active agency*, a term which describes the sense that the robot is behaving and reacting as a living thing would, when provided with interaction stimulus.

The development of this intentionality within the scope of manufacturing, particularly in cases of direct interaction has given rise to the notable concept (recurrent in recent literature) of *trust* between human operators and their robotic teammates. Defined by (Sadrifaridpour *et al.*, 2016), the concept of trust is described as a measure of familiarity and security, based on the performance of both the human and robotic entities, and the rates of successes and failures (Hoff and Bashir, 2015). The model can be time-series dependent, with prior trust levels incorporated into the metric to provide a measure of relationships over time or the concept of Building Trust. This is achieved through increased confidence that the autonomous entity will behave as intended and be of genuine assistance (Heitmeyer and

Leonard, 2015). Additionally, (Lohani *et al.*, 2016) find that inclusion of socioemotional features in robotic teammates promotes the development and subsequent feeling of trust.

As will be expanded in the following section, the scope for variation in human performance is large and is governed by multiple factors, with significant implications for numerous manufacturing tasks. What is evident from the literature covered in this section, is that whilst the potential capabilities that intelligent processing provides have been successfully leveraged in terms of physical interactions, significant work remains to reconcile areas of cognitive processing to enable these same capabilities to improve interactions more passively; such as leveraging adaptability reduce the impact that performance variation has on modelling and process control within the scope of manufacturing systems.

2.5. Human Factors

In the application focused on in this work, these interactions occur between robots and their human colleagues, either directly or as sequential members of a production process. As highlighted in the previous section, the extension and applications of decentralised control to manufacturing processes with the aim on enabling intelligent behaviours necessitates the consideration of the interactions of intelligent agents, both with each other, and their human counterparts. At the system level, this influences the implementation of intelligence and how this may be leveraged to provide *collaborative* behaviours in robotics, and how multiple agents (human or robotic) may observe and adapt to one another to achieve a common goal, is particularly applicable to dealing with the challenge of Human-Robot-Interaction. This is due to the fact that humans possess their own agency, as well as significant variation in their task performance, which is the result of numerous external *human Factors*, most notably fatigue (Hart and Staveland, 1988; Lorist *et al.*, 2000; Lorist *et al.*, 2002). The performance

variation and ultimately disturbance to these manufacturing processes that these factors induce, requiring a degree of adaptability to overcome.

The origins of the study of human factors and how they influence performance and productivity in labour tasks stem from Frederick Taylor's investigations into manufacturing processes at the turn of the 20th century. This work consisted of the analysis of a variety of manual operations, through timings and empirical studies. This study, and the data captured from it, ultimately enabled Taylor to formulate his theory of *Scientific Management* (Taylor, 1911; Taylor, 2004), which forms a collection of best principles to improve the productivity of human workers. The principles of Scientific Management persist to this day (thanks to the efforts of his wife in continuing Taylors work following his death) and have become pervasive throughout manufacturing processes in terms of both design and management. In many ways, it can be considered the work responsible for the introduction of process monitoring, and abstraction of these processes into models and databases; which has continually increased since. It is perhaps, therefore, no coincidence that a study focused on human beings has paved the way for the emergence of many technologies, in particular, Cyber-Physical-Systems (CPS's) based approaches to digitalisation, which are dependent themselves on human data for modelling and monitoring of real-world systems, prediction, and analysis.

There are several factors which exist that are known to influence human performance, leading to decreased repeatability, accuracy, and other variations in performance ability. These factors are the result of multiple different environmental and contextual conditions, and the additional introduced variation means that in contrast to their robotic counterparts, human operators are a source of disturbance to manufacturing systems. This disturbance renders most optimisation techniques ineffective, due to lack of effective mathematical modelling.

In addition to the natural variation between different human individuals, the aforementioned factors exacerbate these differences and have a greater or lesser degree of influence on behaviour and consistency of performance depending on the individual. Many of these factors themselves were identified by Taylor's work, and from a business management perspective, a large body of research exists covering the causes and effects of human factors. This research has been instrumental in modelling the influence of these factors on human operators in the manufacturing context.

The Type of Task and the combination of both mental and physical demands on the individual is a significant source of human performance variability. The NASA developed 'TLX' framework (Hart and Staveland, 1988) identifies multiple types of task, each characterised by different combinations of physical and mental demands, referred to as task demand characteristics. These characteristics are used to categorise different tasks, in terms of how these factors, such as the task duration, influence the perceived workload. Instances of increased workload perception have been associated with correspondingly increased stress, fatigue, and resultant decreased task performance. (Hart and Staveland, 1988; Hancock, Williams and Manning, 1995; Driskell and Salas, 2013).

Of these tasks, assembly operations often require manual and dexterous manipulation of components; in particular, those which remain prevalent within manufacturing. Due to the combination of mental and physical demands, such tasks are very susceptible to the effects of fatigue, which will, in turn, influence task performance. Fatigue is a complex, albeit well-studied phenomenon, and is understood to exist in two distinct types. These are motor fatigue, involving fatigue of the physical biology of the muscle; and cognitive fatigue, which results in the deterioration of cognitive functions (Lorist *et al.*, 2000). The two types also do not occur independently of one another, with relationships understood to exist between the level of motor fatigue and the level of cognitive loading, which results in poorer response

times in decision-making tests, and a decrease in motor control and physical function (Lorist *et al.*, 2002). The relationship between the different types of fatigue supports the assumption that dexterity is detrimentally affected by increased cognitive loading. In addition to the task demands, the time-on-task is another factor which influences the rate of fatigue accumulation and its effects. The cumulative nature of fatigue means that repetitive actions which require physical or cognitive resources will have an increasingly greater effect on task performance (Benedetti *et al.*, 2015). Additional factors, such as how periods of rest are structured and their duration, will influence the fatiguing mechanism (Enoka and Duchateau, 2016).

The use of the term fatigue to describe this depletion of resources leads to the discussion of the more traditional understanding of the term, whereby fatigue is used to describe tiredness and the effects of sleep deprivation. Sleep is the primary mechanism by which physical and cognitive resources may be replenished, reducing the accumulated fatigue before rest. Consequently, it can be seen how a lack of sleep effects fatigue, and that both the immediate and cumulative effects of sleep deprivation have a significant influence on performance (Koslowsky and Babkoff, 1992; Dinges *et al.*, 1997). Sleep itself, as a biological phenomenon, is at best, poorly understood. What is known, is that Human sleep patterns are governed by natural cycles known as circadian rhythms, which dictate in terms of physiological activity, periods of activity linked with increased motivation and task performance. These cycles are understood to be closely linked to the time of day (Blake, 1967), and the study of these circadian rhythms has identified several patterns. These patterns are termed chronotypes and are often expressed through a preference for either morning (larks) or night (owls) activity. This preference, as a result of differences in physiological activity, corresponds to decreases in task performance at a non-preferential time of day (Kerkhof, 1985). In addition to the time of day, there are several additional factors which influence these periods of increased activity. These include the quality and intensity of

environmental light levels, with better illumination associated with improved task performance (Campbell and Dawson, 1990) and by satiety, which is both susceptible to, and determinant of, many of the bodies same physiological processes (Berthoud, 2007; Lowden *et al.*, 2010). Other studies have also led to the observation of a day-of-the-week effect, with decreased performance on Mondays, rising through the week to optimal performance on Thursdays (Testu and Clarisse, 1999), however, it is not known whether this phenomenon is associated with fatigue or is social in nature.

The factors detailed in this section represent a selection of the most well understood and most influential in terms of their effect on task performance. The truth of the matter is, that the human machine is governed by complex interwoven processes, and the number of potential sources of variation is unquantifiable.

2.6. Summary

The exploration of current literature identifies several trends. Under international initiatives, such as Industry 4.0, development of numerous enabling technologies has been accelerated, in particular, those relevant within in the context of intelligent manufacturing. As human beings remain an integral part of many manufacturing operations, the increasing demand for autonomous, reconfigurable, and adaptable robotic elements within these manufacturing systems has led to interactions, and hence collaboration, between humans and robots becoming increasingly prevalent in the manufacturing context. This presents issues in terms of optimising these systems, and sets limits on potential productivity, due to the disturbance introduced into these systems by variations in human performance, both as individuals and as a result of other human factors. The unpredictable and inconsistent nature of this disturbance has an impact on the processes ability to operate in a lean manner resulting in tangible increases in observed idle times, costs associated with WIP, and overall productivity.

The preceding section has explored a wide area of literature from various fields, which, in line with the proposed research questions, suggests that there are opportunities for the increase in the capacity of computers to act intelligently to improve the interactions between robotic operators and their human collaborators; and reduce the impact of human task performance variation on manufacturing processes. This section aims to quantify the existing body of literature with respect to its relevance to the main research questions. This includes identification of the areas in the existing research which are lacking, and the questions that this poses in terms of advancing work in this area. Within the scope of this work, this relates to how the techniques promised by *intelligent manufacturing* may be applied within the manufacturing environment, so as to facilitate the interactions between human and robotic operators and to understand and alleviate the consequences of the disruption introduced by the presence of human beings.

With respect to the first of the main research questions, this involves work on the variation and adaptability of human beings and the consequences this has for manufacturing processes. This has been well studied in many direct domains and physical interactions, and novel ways of leveraging new technologies and methods to eliminate the negative effects, and leverage for the better, this variability, have been developed. There remains, however, limited work into how these same traits influence more passive interactions, such as the system-level dynamics of these processes, and the effects. There is also a lack of work focusing on applying these new technologies based on principles of adaptability to working with human beings in such a way.

Answering this question will build on existing research on social-intelligence and its application to autonomous systems and how they interact with human beings. In terms of furthering social intelligence, the act of collaboration requires effective perception and communication of another, to coordinate action to achieve a goal; and the problems presented

in terms of performance disparity provide the perfect case to explore the application of these concepts. From a process management perspective, it will provide a solution to human modelling, and address the application of social intelligence to adaptive control systems allow real-time changes in behaviour, in response to human variations. This will be achieved through enabling robotic operators to understand the variable behaviours of individual human operators, and to take the appropriate actions, resulting in more adaptable behaviours and decision making, at the social level.

The second of the identified research questions seeks to provide insight into how this intelligence may be implemented and realised within these systems. From the existing research, this intelligence is typically manifest as the product of analysis of observed data by learning algorithms, which extract information from representations of the system or environment they are trying to model. This information is then used to make predictions about system variables or to determine appropriate actions through some selection policy. What is clear from the existing literature, is that the scope of these algorithms is vast, and there exists no definitive guide as to the applicability of different approaches. Neural network-based techniques have emerged as a versatile methodology for abstracting the relationships between data inputs and outputs and have been applied successfully in a variety of analysis and control applications; and there are significant parallels between neural network-based approaches and cognitive theory. As such, through a neural-network-based approach to analysis, this work aims to provide insight into how these techniques may be used within the context of facilitating human-robot-interactions, both in terms of understanding the dynamics of the system and for analysis and decision-making. This requires consideration of how to capture the relevant representation from the environment, and how this is presented to these algorithms, to provide effective learning, development, and evaluation; which is crucial to facilitating effective human modelling.

The last of the research questions seek to reconcile the numerous relevant domains within intelligent manufacturing, and the existing work further suggests that the realisation of intelligent systems requires the identification and evaluation of methodologies for the effective combination of numerous techniques within this domain. These techniques include: digitalisation, to functionally, yet representatively model interactions and control processes, distributed control, to enable appropriate processing of observed information; and machine learning techniques, including the application of the identified reinforcement methods to identify self-optimising behaviours within the modelled application; which must all work in synergistically to achieve the resultant desired behaviours.

This is evidence by the advent of Multi-Agent-Systems, which address directly this problem of coordinated and decentralised control. Combined with the capacity for intelligent behaviour, these agents may be designed to coordinate their actions with observations of their collaborators, to improve the interaction between them. This has been demonstrated already in many applications that are reliant of physical interactions, but there is a lack of work on exploiting these capabilities to improve these relationships at the passive level; and how it may be applied in both theory and practice. It is unclear is how such opportunities can be capitalised upon, and there exists no agreed-upon method for realising the theoretical benefits of intelligent systems able to achieve a level of social cognition. This is due in part to the relative infancy of the technologies and methods involved, but primarily the result of the wide variety of applications. As a consequence, any developed systems should aim to generalise these interactions whilst maintaining representative accuracy. Much of the existing work is outdated, however, and the capability of modern systems far exceeds those for which these frameworks were designed. Despite this many of the insights and methods on how autonomy and adaptability can be replicated through the use of cognitive computing and multi-agent-systems remain useful.

Establishing these methods and realising a cohesive application of the identified and developed techniques in such a way as to be applicable to a real-world process, is a key aim of this work, and resolves many of the gaps in the existing research regarding how intelligent methodologies may be applied to the problem of human variation. Similarly to the first case, many of these technologies have been explored within similar applications, but the use of these methods to improve the interaction dynamics in the care of human-robot-interaction has yet to be explored, despite the very realisable benefits at the system level, in terms of reductions in observed idle times, increased productivity, and optimisation of the process through improvements to its lean operation.

The research suggests that the implementation of intelligent agents may leverage several benefits of decentralised control and machine learning techniques, to enable robotic operators to behave in an adaptive and crucially *social* manner concerning their human counterparts. Adaptability within the robotic elements of manufacturing systems has begun to improve human-robot-interactions in the physical domain, however, despite the ubiquity of human workers and the obvious disruption that is associated with human performance from a production management process at the system level, it is an area that remains to be addressed.

The benefits that adaptation can provide within the manufacturing setting are clear from the existing work, and consequently, it is relevant to explore the application of these methods to the problem of human performance variation and more passive interaction behaviours. Enabling *social* behaviours, which are characterised by benefitting both agents involved, is an example of how these passive interactions may be leveraged and necessitates the development of agents capable of social cognition. The following section outlines the research framework used to implement the necessary elements of cognition, and how these relate to enabling robotic operators to adapt their behaviours as a response to their human collaborators and facilitate these interactions, on both a human and process level.

3. Research Framework & Methods

This section covers the developed research approach to enable the implementation of intelligent agents within the context of manufacturing control. The research approach is designed around answering the key research questions, and the key considerations were therefore development of a suitable modelling approach to enable interaction between the software and hardware components, and how the collection and processing of data relates to the domain of manufacturing systems. The first chapter of this section covers the definition of a framework, which is concerned with the handling of information within such systems and highlights the necessary elements for consideration to realised desired behaviours. In addition to the existing literature, the research framework is also informed by the processes and systems in place in the manufacturing processes of the industrial partner for this work, namely the separation required between the elements of data collection and robotic action execution. This separation provides the opportunity for interstitial data processing, and clearly separable external boundaries for the intelligent agent.

The second section in this chapter further discusses the key framework elements in terms of the relevant technical and research methods to be used. These methods were again defined to best answer the proposed research questions, name monitoring processes used by the industrial partner also served to inform much of the approach towards the necessary elements of processing, both in terms of how the data was captured formatted and stored, and the necessary processing steps required to retrieve actionable information from it. This includes elements like the ability for the agent to execute learning and complex analysis separately from decision making to facilitate software embedding, to pre-process and reformat incoming observations, and to execute output actions independently of the application platform through the use of appropriate software API's.

3.1. Research Framework

As established from the literature, there is good evidence to support the notion that enabling *intelligence* within manufacturing systems through the use of learning and decentralisation can facilitate adaptable behaviour of robotic operators. The decentralisation of control systems from the hierarchal level (where a central processor is used to monitor and control the parameters and execution of the entire process) to individual robotic operators (through the use of learning to enable an intelligent analysis of their observations), can improve collaborative behaviour, through the appropriate selection of action based on the observed state of the process and collaborators. This framework established the approach to this problem from the perspective of facilitating this collaborative behaviour, intending to minimise the impact of human task performance variation on the process.

There are two independent disciplines which can be seen in a contemporary automated manufacturing process. These are the disciplines of *Data collection*, which in this context broadly describes the generation, collection, transfer and storage of data. And the discipline of *Robotics*, which is responsible for the hardware elements of automation, including the robotic manipulators and controllers themselves, and the connection of these physical elements to virtual systems.

Whilst these disciplines exist separately, as data-driven robotics systems become more commonplace, the separation becomes less distinct, and there is an increasing need to formalise the interaction between the generated data, its representations, and the physical system elements. In these data-driven systems, (typically binary) signals from sensors in the data collection domain, are communicated to a robotic controller (Such as a Programmable Logic Controller PLC), in the robotics domain. This signal will trigger the appropriate response logic and corresponding action. The flow of information in this respect can be seen in the framework illustrated in Figure 3.

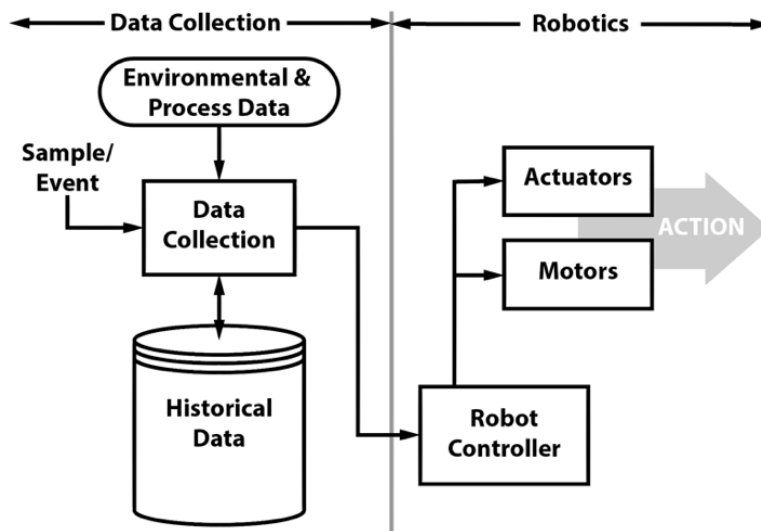


Figure.3 - Existing information processing systems illustrated in terms of information flow through the system.

The data generated within a sample of a manufacturing process can be defined as being from one of two sources: *Process data*; which are the data points related directly related to the elements and parameters of the process; and *Environmental data*; any supplementary data providing contextual information, which may be analysed for additional insight. These sources of data may also be supported by databases of historical data, in certain applications.

Whilst these signals can be centrally collected, processed, and more appropriate commands sent back to these controllers, as discussed, problems with these hierarchal systems become apparent when dealing complex systems (containing many individual machines, each with potentially thousands of sensors). Consequently, to enable the distributed intelligent processing of this data, the following framework is proposed, which outlines the necessary constituent elements of processing required, and the interactions between these elements; to enable adaptability and facilitate collaborative behaviours.

The framework in Figure.4 illustrates the addition of an intermediary *Cognitive Layer* which contains these modular elements to implement the necessary data processing steps. This cognitive layer is software-based and can be implemented on-board a robotic operator to enable an *intelligent* response to changes in the perceived environment, to provide agency in the robotic systems, and facilitate interactions with their human counterparts.

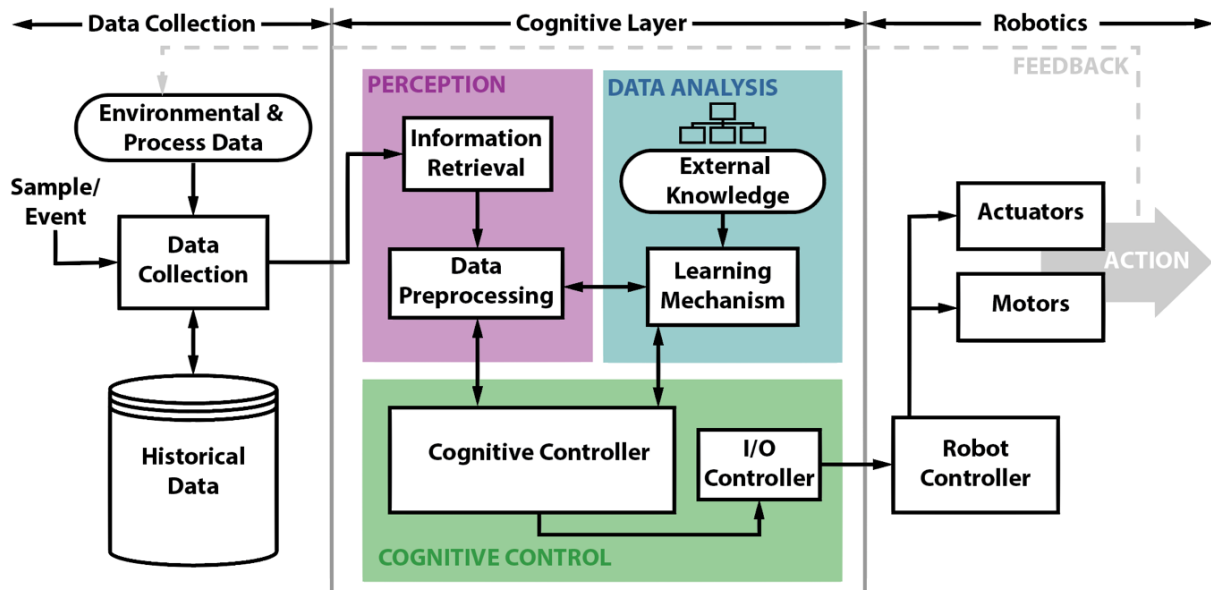


Figure 4 - The proposed framework illustrated in terms of information flow through the system. Divided into three functional layers.

As can be seen from Figure.4, there are three distinct areas of processing that are defined to enable the necessary data processing and end behaviours in the intermediary the proposed *cognitive layer*. The layer is based on the modular structure seen in existing cognitive architectures reviewed in the literature, which is leveraged here as the isolation of high-level cognitive processes facilitates integration with low-level control. Each of the three modules combines is responsible for a different area of processing of the data and leverage multiple combined functions to replicate different cognitive processes and enable appropriate behaviour to be selected. The following section outlines each of these modules in more detail and discusses the technical application of these processes, and the approach considered in this work.

3.2. Research Methods

Within the developed research framework, there are several distinct areas of processing, each of which must be suitably approached from the perspectives of both development and evaluation. The *Perception* module is responsible for providing the interface between the software agent and the data collection system. This is in terms of both capturing and appropriately and pre-processing relevant data in a manner that may be utilised by the other modules. Perception is a more accurate term here than observation, as the observed data and the information it contains is affected by the beliefs and aims of the observer in terms of how this data is processed. This may be formatting, or the derivation of more useful data instances, for example, establishing a cycle time by looking at the timestamp separation of execution of two different sensor activations. Additionally, the methodology for data collection is application-specific and dependent on the systems implemented. The framework identifies the critical elements that the data collection system must be able to capture and recall. Minimally, this is the capacity to gather, store and transfer multiple data instances, both real-time and historical, in a format which may be parsed by the software agent.

The *analytics* module isolates the learning and analytical processing of the cognitive layer and is responsible for higher-level, analytical processing, focused on extracting information from contextual data. The analytics is done by utilising a neural network to build a predictive model to approximate the relationship between observed data and the desired output value. The module also provides storage to supplement the observed data with additional information that is not observable by the agent. This may include additional contextual information, such as shift patterns or production targets. Use of machine learning techniques for knowledge discovery here include consideration of multiple factors, which as mentioned are frequently application specific. Necessary consideration must be given to the data and format, including relevant pre-processing steps and suitable evaluation metrics to

asses learning and predictive performance. Consequently, specifics of the methods used for the realisation of these functionalities are detailed as part of the development of the intelligent agent.

The *Perception* and *Analytics* modules provide support to the *Cognitive Controller*, which manages the information flow through the cognitive layer, and is responsible for enacting the actions of the agent. The internal structure of the module and its relationship to the previous two is detailed in Figure.5.

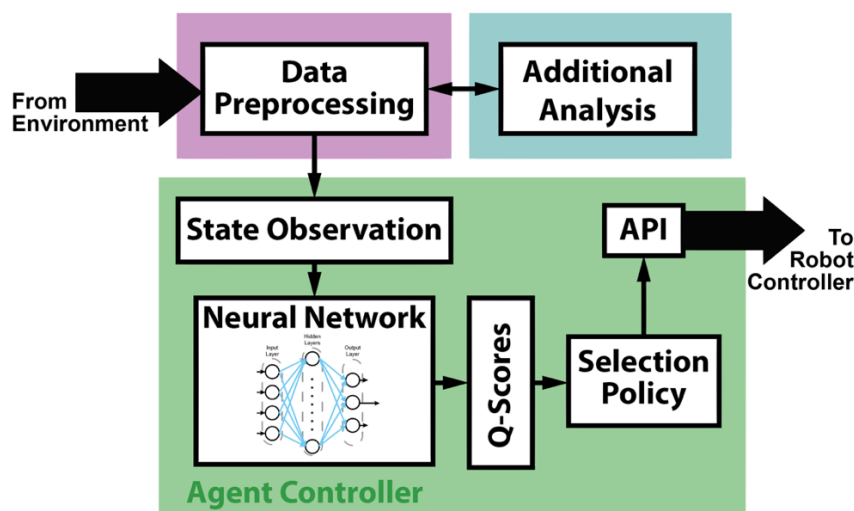


Figure 5 - Architecture illustrating information flow through the Agent, and how it interacts with the simulation environment.

The cognitive controller contains the functionality to implement Q-Learning, following the process of an MDP. This enables the agent to optimise and select the operations and parameters of the robot to exhibit the necessary functionality, based on the information provided by the other modules. In addition to selecting the appropriate actions, the cognitive controller is also responsible for issuing the relevant commands to the robotics hardware, through the use of the relevant API's, where they can be enacted using existing control techniques. In many robotics applications, the instructions will be received by a Programmable Logic Controller (PLC), is then responsible for generating the necessary command signal for the motors and actuators to affect the relevant motion of the robot.

The overall design of this cognitive layer is designed to act as a software agent, capable of leveraging Q-Learning through an MDP. Consequently, the design is based around developing an observation which describes the environment; performing the algorithmic and learning aspects of the process, resulting in an *action*, which influences the system, in turn, recorded environmental and process data, forming a feedback loop of the next state representation.

The clear distinction between the *digital* and *physical* domains of the system facilitates the division of cognition as a whole into the constituent processes associated with higher-level reasoning, reducing the computational load of the processing; whilst isolating the processes responsible for *control planning* which ensures correct motion of the robotic end effectors etc. (motor control being a separate area of cognition entirely (Lorist *et al.*, 2002). This preserves *reactive* action as control can still be triggered by sensors connected to the robot controller located directly in the process (i.e. kill-switches and collision/fault-detection). Using established automation techniques will also facilitate implementation and compatibility in terms of realising such systems.

The architecture presented in this section illustrates the design and implementation of an intermediary *cognitive layer*, to provide robotic operators with adaptive functionality, based on the knowledge extracted from real-time contextual information observed from the environment and collaborators. The following section outlines the methodology to develop the necessary software elements, to enable the inclusion of knowledge of human factors and utilise adaptable behaviours to account for the variation and uncertainty resulting in the interactions with human beings and consequently these production processes. As discussed in the literature review, the use of q-learning through a Markov process has a significant overlap in compatibility with discrete event simulation (DES), providing distinct, discrete states, and

actions to transition between them, as such, a DES approach to explore the application of these machine learning techniques is a logical consideration.

The following three chapters present the methodology developed to identify and overcome the demands and considerations required to create an adaptable control structure as outlined in the previous section. This methodology consists of several individual sections, each concerned with the development of a different aspect of the approach. These are **Human Factors Modelling & Simulation Design**, which is concerned with the effective modelling and parameterisation of human elements and human factors influence within the simulation model, the capture of human performance data, and the development of a simulation model in such a way that it is generalised to multiple real-world scenarios and provides the necessary functionality to explore the areas of reinforcement learning and human-robot-interaction; **Agent Design**, which covers the development of how the agent interacts with the simulation environment, how the environment is captured and provided to the agent, the actions necessary to achieve the relevant adaptable behaviours, and how the reinforcement aspect is realised through rewards and software design; and **Agent Evaluation**, which details the approach used to evaluate the performance of both the neural networks and the combined intelligent software agent.

4. Human Factors Modelling & Simulation

This chapter outlines the development of an effective methodology to model and simulate interactions between robotic operators and their human counterparts, as they occur passively in the scenario observed in a variety of manufacturing applications. The chapter covers two main areas, the first relating to the dynamics of these interactions as they occur in these processes, and the development and application of methods to effectively model and consequently simulate these interactions in such a way that the interaction is generalisable to any specific production process. The second key area relates to capturing and understanding the influence of human performance variation on these interactions, and how this may also be effectively captured, modelled and simulated, in such a way that is representative and generalisable to as many of these interactions as possible. Furthermore, this second section also explores the developed methodology for the effective parameterisation and modelling of human task performance, based on the monitoring of individuals over a sample period. This has particular relevance in terms of both improving the general validity of the approach, but longer-term consequences, which will be discussed as appropriate.

4.1. Simulation Design

The initial consideration is the appropriate digitalisation of the production process and relevant interaction. A simulated version of a manufacturing process, that is suitably representative, is crucial to enable the effective development of a software agent. This scenario must be appropriately generalised in order to improve the validity and reliability of using a Q-Learning approach in such a way.

A process consists of multiple operations, which -in an automated context- are performed by several robotic operators which together constitute the production line. The simplest production line model consists of a direction of motion, and the location of

operators who perform manipulation of the product at some point along the path of the product. These manufacturing *cells* are separated by conveyors for transportation and frequently have a fixed capacity. Within these operations, there may be several cells where tasks are completed by human operators. Despite good design and optimisation, this presence of human operators remains a source of disturbance, due to their variation in their respective cycle times, which introduces a performance *disparity*. This forms an interaction as illustrated in Figure.6:

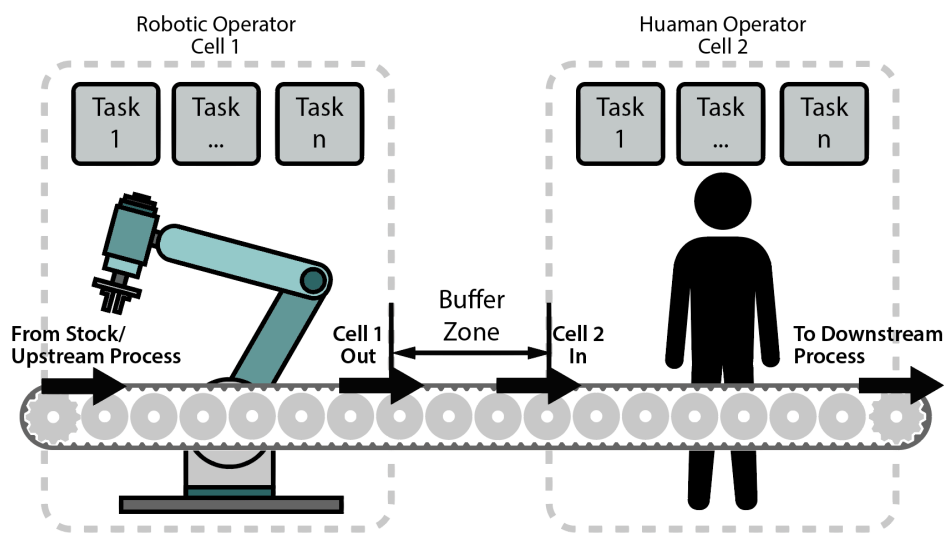


Figure 6 - The model of interaction between human and robotic operators typically observed in a manufacturing process, each cell has an associated Cycle Time.

To illustrate how this disparity has implications for the overall efficiency of these interactions, assume there are no upstream constraints, to fulfil his goal (in the manufacturing context, the simplest goal definition is to maximise the volume of production) with no consideration for the downstream effect of his actions, the operator should logically attempt to complete each cycle as fast as possible. However, even simple optimisations will often reveal that working at capacity is not necessarily the most productive course of action. Where there is a disparity in performance which is allowed to persist over time, there occurs either a large surplus or deficit of products between the two operators, as the products completed by

operator one builds up, or fail to be delivered in a timely enough manner. Eventually, this process bottleneck leads to idle time being observed in the system, either in the upstream operator, where the buffer zone becomes filled and they must wait for space to become available and in the downstream operator, where they must wait for the next product to be delivered. In the former case, the disparity also introduces an increase of *workpieces in production* (WIP). Therefore, to improve these interactions that arise from the design of production systems in this manner requires each operator to consider the conditions at other locations and the actions of his collaborators. Considering the upstream operator, conclusions about how to adapt behaviour in the social context become apparent. Given an awareness of the downstream operator, the upstream operator should be able to tailor their behaviour based on these actions, to optimise their work, and consequently, the process as a whole. The formulation of the interaction in this way bears similarity to fetch-and-deliver type interactions where one agent must provide the other with an object for them to perform their task. These have been studied in the field of human-machine interaction, often in terms of *anticipation*, and are defined by their *fluency*, which is a term used to describe the disparity in time between the need of the collaborator and the response of the agent. A lower disparity equivalent to higher fluency, and *better* interaction (Hoffman, 2013).

To enable exploration, a simulated environment was developed using the AnyLogic simulation platform (PWC, 2018), a Java-Based software package designed for Agent-based, Discrete Event and System Dynamics simulation approaches. The software package provides several advantages to the applications of this work, including functionality for different modelling approaches, and the ease of extension and integration with external Java Libraries and custom software objects. This simulation environment was designed to replicate the identified model of interaction between a Robotic Operator (RO) and a Human Operator (HO), performing product assembly tasks as part of a production line. As far as possible, the

simulation is designed in a generalised manner, to provide a basis for exploration of how different actions may be determined, selected, and enacted to improve the fluency of these interactions. This is achieved within the simulation, by discretising specific sub-operations into a reconfigurable set of logical elements, which enables the methodology to be applied to a non-specific manufacturing operation. Within the model, combinations of these elements form the human and robotic agents, which are able to be defined by the task duration or Cycle Time (CT). Initially, the interaction dynamics are modelled in terms of an upstream and downstream position, with the cells of each operator are separated by a conveyor (doubling as a buffer zone). This follows the interaction typically seen in production processes identified in Figure.6 previously, and the model itself at the top level is illustrated in Figure.7.

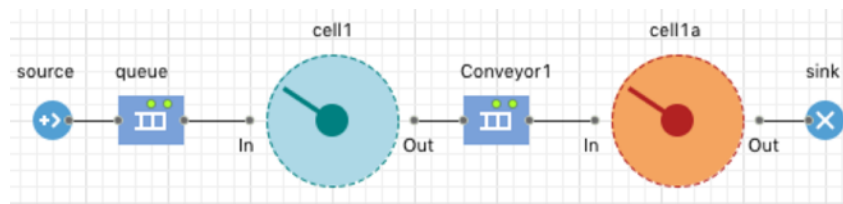


Figure 7 - The model developed and the corresponding Anylogic simulation, each cell contains a delay and data capture function.

Developing a machine learning model to of the relationships between the observed information and the resulting performance, in terms of both prediction of the impact of human factors, and in terms of reinforcement, is a key aim of this research. The literature has identified the advantages of a neural network approach, in particular, due to their excellent generalisability as function approximators. The robotic agent in the simulation model can be said to represent the hardware aspect, within this agent, a software object is developed to provide the neural network functionality. This development was done using the Java-based DeepLearning4j (DL4J)(D14j, 2018) to facilitate integration with the Java-based simulation platform; The Java classes which define the Neural Network behaviour are packaged using

Maven to produce a Java Archive file (.jar), which is included in the AnyLogic model as a Dependency enabling access to code developed using the DL4j library. The integration of the neural networks in this way enables function calls to be made to the software agent during simulation execution, enabling training and evaluation in a custom, reconfigurable, and dynamic task environment. Two of these neural networks are defined, each within their own modular code, to represent the different cognitive modules. The first is used in the analytics module and performs the multidimensional regression to provide a prediction based on an observed instance, of the collaborators' cycle time. This value can then be used to form the state representation, enabling decisions to be made based on knowledge of collaborator performance without direct observation, and at different points in time. The second neural network forms the reinforcement learning system which governs decision-making with respect to the state observation, to reduce the disparity in performances, resulting in a reduction of the number of Workpiece-In-Progress (WIP), and the idle time observed for either operator, consequently improving the fluency of the interaction.

In addition to evaluating the accuracy of the neural network during training, integration of the learning element into the simulation environment is necessary to evaluate the validity of the approach and performance of the intelligent agent in terms of a representative (albeit generalised) task. This enables a more accurate assessment of the developed model in terms of how well such an approach can be used for real-time adaptive control. Developing the functionality to enable intelligent processing and adaptable behaviour is key to realising the potential of intelligent manufacturing systems and realising an intelligent agent which can provide accurate predictions based on observation will contribute to further understanding of the suitability of the approach.

Furthermore, the presented method enables the development of a suitably representative simulation environment to be built following the presented principles of

generalisation, if greater fidelity is needed, which may be used to provide a training and evaluation platform for the development of these agents, in a customisable and application-specific manner. Additionally, the use of the simulated environment enables this in a manner which is non-invasive to the real-world process. These are both strengths of the approach, facilitating the development of optimal solutions with minimal disruption, through ease of iterative evaluation.

4.2. Human-Factors Modelling

A crucial element in designing the simulation environment is the modelling of human task performance in such a way as to effectively model the influence of human factors, whilst remaining representative. The simulation acts as a platform to evaluate *on-task* robotic performance but is also used to generate data for training the learning algorithms. For both of these reasons, appropriate care must be given to how this data is calculated and captured. To achieve this, the components of the simulation representing human operators are parameterised to replicate both the variation between different operators and the influence of human factors on their performance. As discussed in the literature review, the majority of human factors are expressed through or are directly related to the mechanisms governing fatigue; which in turn has the most significant influence on human performance across almost all types of task.

To achieve this, several variables were defined representing different aspects of fatigue, which were then used to adjust the human task performance during the simulations. This was achieved by modifying the nominal Cycle Time (calculated as the total time duration between products leaving each cell) which is used to quantify task performance, as discussed in the previous section. The cycle time is directly related to the value of a *delay* element within the human agent, with a variable time value, which is manipulated through the use of these

modifiers to represent their influence. The modifiers were selected, as mentioned, based on factors influencing fatigue, which will influence individuals to a varying degree. As such the values of each modifier can be defined to replicate a variety of influence and susceptibility between individuals. Furthermore, the influence of several environmental effects including Noise, vibration, and light-levels, were not considered in the model, as their impact on performance is comparably negligible if they remain consistent.

Three different profiles are defined to provide a wide range of conditions and combinations of susceptibility to the impact of the identified human factors. Operator 1 was parameterised as an experienced operator, with a low initial cycle time, but susceptibility to time-on-task fatigue; Operator 2 was designed to represent an average case, with a nominal base CT equal to that of the designed cycle time for the system processes (or *takt time*), and no fatigue influence was included; Operator 3 was designed to represent a more novice worker with a slower cycle time, but a pace which reduces the influence of fatiguing. The effects of fatigue on task performance are modelled with the use of a *fatigue modifier*, which assumes a linear evolution over the shift duration is calculated by scaling the maximum effect over the elapsed shift duration, as per Equation.2.

$$Fatigue\ Modifier = \left(\frac{Elapsed\ SD}{Total\ SD} \right) \times \left(1 + \frac{\%Increase}{100} \right) \quad (2)$$

The human elements are additionally parameterised to represent the influence of the time of day, through a *shift modifier*. Operators 1 and 3 were considered *Owls* and receive a penalty of a 10% increase in cycle time during morning shifts. Similarly, the day-of-the-week effect is represented by a *weekday modifier* for Operators 2 and 3, representing their particular susceptibility, and increasing the individual variation. This modifier was set to decrease performance at the start of the week and shift this influence to a 10% performance increase on the penultimate day, replicating observed patterns. This combination of the modifiers enables the calculated cycle time for each human operator at a specific point in

time (Monday, AM, in the illustrated example.) to be obtained via equation.3, where n is the specific operator, and the values for each modifier for each defined operator is illustrated in Table.1., where, WM, SM, and FM, are the corresponding weekday, shift and calculated fatigue modifiers respectively.

$$\text{Calculated } CT_{amn} = CT_n \cdot WM_{an} \cdot SM_n \cdot FM_n \quad (3)$$

Table 1 - Breakdown of the values used to modify the performance of each operator.

Operator Number	1	2	3
Base Cycle Time	CT_1	CT_2	CT_3
Fatigue Modifier (End of shift)	FM_1	FM_2	FM_3
Shift Modifier AM	SM_{AM1}	SM_{AM2}	SM_{AM1}
Shift Modifier Middy	SM_{MD1}	SM_{MD2}	SM_{MD1}
Shift Modifier PM	SM_{PM1}	SM_{PM2}	SM_{PM1}
Weekday Modifier Monday	WM_{a1}		WM_{a2}
Weekday Modifier Tuesday	WM_{b1}		WM_{b2}
Weekday Modifier Wednesday	WM_{c1}		WM_{c2}
Weekday Modifier Thursday	WM_{d1}		WM_{d2}
Weekday Modifier Friday	WM_{e1}		WM_{e2}

Parameterising the simulation model with these modifiers and through the outlined equations, enables the generation of models of HO performance, to induce different interaction dynamics and varied conditions. This is done by iterating the day of the week (Monday-Friday), and the shift order of the operators each week, to vary the Time of Day. Each simulation run represents one day of operation, and consists of three shifts, am, midday, and pm, each performed by a different human operator. The operator assigned to each shift is then varied week to week, to represent the performance of each operator across the full range of working conditions (These orders are: 123, 231, 312). This was done for a total of fifteen simulation runs, with every 5 simulation runs representing a working week. This has

relevance for two reasons, Firstly, it enables the generation of training datasets which profile the performance of different operators under the influence of varying combinations of these factors. This enables the neural network to infer and predict, from these four contextual input features, the Operator Number (ON), Shift Number (SN), Shift Duration (SD), and Weekday (WD) values (and the corresponding cycle time (CT) as the label), the impact on task performance of the HO, based on historical performance. Additionally, aggregating the performance data for each operator into one singular dataset in this way facilitates the discovery of patterns in performance that are independent of an individual HO or which exist over different timescales. Combined, these values form the *observation space* of the analytical network, and consequently must be included in the observations supplied to the intelligent agent, which will be discussed in detail, in a subsequent section.

Additionally, where these values take a categorical type, i.e. in the case of the shift number and weekday attributes, it is important to consider their presentation to the agent in terms of a *one-hot-encoding*. Representation of categorical data points in a one-hot encoding dates back to the early days of computing, as a technique for efficient memory management, but has significant applications in terms of improving the learning efficacy of neural network approaches. It does this by separating the possible values into separate binary inputs, removing the influence of scaling; as the magnitude of the value is not important in these cases, only that it is distinct. To demonstrate, the weekday variable takes a single input value of [4] on Thursday, which becomes [0,0,0,1,0], under a one-hot encoding.

4.2.1. Human Data Profiling

For use within the simulation model, it is also necessary to consider the randomness of individual human beings, and how their lack of repeatability may influence the ability of the agent to develop an appropriate policy. Consequently, a methodology must be developed to

appropriately profile human individuals, to enable effective modelling of these individuals within a typical manufacturing scenario, and without the requirement for constant surveillance of other data-driven approaches. The methodology proposed aims to make use of a period of initial monitoring, from which, a representative set of parameters can be calculated, and used to represent the variable behaviour and task performance of individuals within a simulated process. In this instance, the task performance is evaluated using the cycle time, and this value will be used to build these profiles. In addition, the methodology aims to capture the variation in this metric along several vectors of variation.

To parameterise variability, the initial monitoring period makes timings of a relevant number of manufacturing cycles for each of the individual operators. These values can be averaged and used to parameterise a probability distribution, which in turn sampled, and used to affect the cycle times of the human operators modelled in the simulation environment. This allows the behaviours of individuals to be more accurately modelled, and more accurate predictions to be made and provide a data-driven approach without continuous monitoring. It is additionally necessary to establish the appropriate distribution to describe the task variation. An event such as this with a continuous range is often modelled using a normal distribution, centred around a mean value, and parameterised by a standard deviation, σ . However, for an optimised manufacturing process, it is likely that the lower bound for performance much harder than the upper bound, as the number of factors which may slow down process far outweighs those which will speed it up. As such, it is there may exist a skew in the probability distribution towards the faster end. (i.e. more observations will be made of a well-optimised process working efficiently than will be made of events leading to disturbances). As such, the distribution of task performance may be more appropriately modelled as a skewed distribution, or a two-parameter distribution, such as a gamma distribution, to incorporate a shape parameter, k , to define this effect appropriately. This has a

potential consequence in terms of how performance for the operators is modelled and use of an incorrect distribution may misrepresent the observed behaviours when simulated. As such, establishing the appropriate distribution of cycle time variability is critical for effective human modelling. To do this, the distribution of samples for each of the operators is visualised and used to determine the nature of the observed variation in observed cycle times. Once the appropriate distribution has been identified, the appropriate parameters can be calculated for each operator from the collected data, and an appropriate distribution may be built and sampled to provide estimates for human performance within the simulation environment. The theoretical aggregation and parameterisation of this data for three hypothetical individuals, X , Y , and Z , for n samples, is illustrated in Table.2:

Table 2 - Theoretical values measured during observation to enable profiling of performance for multiple human operators, over different conditions; assuming a Normal Distribution.

<i>Operator</i>	<i>Condition</i>	<i>Observation (s)</i>			Mean, μ (s)	Standard Deviation, σ (s)
		<i>i</i>	...	<i>n</i>		
X	c_1	x_{ci}	$x_{c\dots}$	x_{cn}	$\frac{1}{n \cdot k} \sum_{i=0}^n x_{ci}$	$\sqrt{\frac{\sum (x_{ci} - \mu_X)^2}{n - 1}}$
	c_{\dots}					
	c_k					
Y	c_1	y_i	y_{\dots}	y_n	$\frac{1}{n \cdot k} \sum_{i=0}^n y_{ci}$	$\sqrt{\frac{\sum (y_{ci} - \mu_Y)^2}{n - 1}}$
	c_{\dots}					
	c_k					
Z	c_1	z_i	z_{\dots}	z_n	$\frac{1}{n \cdot k} \sum_{i=0}^n z_{ci}$	$\sqrt{\frac{\sum (z_{ci} - \mu_Z)^2}{n - 1}}$
	c_{\dots}					
	c_k					

In order to adequately capture variations in human performance over different contextual conditions, monitoring must also be done over a variable set of timescales. This will enable the adequate parameterisation of digital manufacturing models in terms of each of the data's *dimensions* to reflect the influence of different human factors. Either by using the observations to validate the current models or by directly adjusting the distributions used based on different aggregations of the data.

5. Agent Design for Effective Reinforcement Learning

This chapter is concerned with the design of the software agent in terms of enabling adaptability to balance the operational demands of the system with the demands of improving the interaction with other operators, both robotic and human. The design of this agent covers aspects of how the agent observes and interacts with its environment, from both the perspective of the simulation and the agent itself. It also necessitates the delivery of rewards to the agent and the development of appropriate policy for doing so. Each of the following sections presents the methodology used to establish these design parameters, and how they relate to implementing adaptable behaviour of the agent.

5.1. Agent Interaction

With the capabilities of the intelligent agent defined to enable the reinforcement network to appropriately observe, respond, and affect the environment, the simulation model can be designed appropriately. Using the Anylogic simulation package (Company, 2018), a discrete event-based simulation can be combined with agent-based control to explore agent performance. The package is Java-based, enabling easy integration with the Deep-Learning-4-Java (DL4J library) (DL4j, 2018). The library facilitates the development of a Deep Q-Learning Network, that can be integrated effectively with the simulated environment to provide control decisions. The discrete event simulation model as described in section 4.1 contains a software agent which contains elements to represent several different operations of the manufacturing task. Within a manufacturing operation, we can define a number of actions which enable manipulation of products to complete a process. These are: Pick up and put down products; Move products from one location to another; Capacity to scrap a product in the event of error; and the ability to manipulate products by following pre-set motions/routines.

The elements within the simulation are consequently chosen to reflect the modelling of these abilities and other elements within a manufacturing process and to suitably generalise different variants of these operations based on their characteristics. Queue elements are used to represent individual or groups of products at a given position, such as in a stock of parts, or the products contained in an interstitial buffer; Delay elements are used to represent operations with a definable cycle time (either static or variable) associated with them (i.e. a fixing/glueing operation) and hold the product object for a duration equivalent to this value. Also, a queue element with a capacity of one is used to represent the robotic gripper within this agent, which is used to distribute products around the operations of the process, through variable output gates. These abstracted elements can be combined to represent any number of manufacturing tasks, through generalising different operations and enable the relevant observation and action spaces to be defined for the Deep Q-learning model.

The actions selected by the agent enables a large variety of different commands to be represented dependent on the application, as they are issued as commands to traditional robotic controllers, to follow defined subroutines. Consideration must be given the output actions, and how they influence the simulation environment to enable the desired control to be realised. Within the simulation model, the robotic operator is controlled by a statechart, which makes the appropriate function calls to the neural network and manipulates the simulation through different states that the network selects, each representing the call to a different action subroutine. This enables the control loop to effectively follow the MDP, as it is designed in terms of the definition of a state and response to trigger an action. The generalised format enables the definition of a set of actions for each task, facilitating the application of the approach, and isolates the interaction of the software agent from the rest of the simulated environment, whilst providing an efficient channel of communication, for

structuring events as they occur within the MDP. The statechart takes the generic form illustrated in Figure.8.

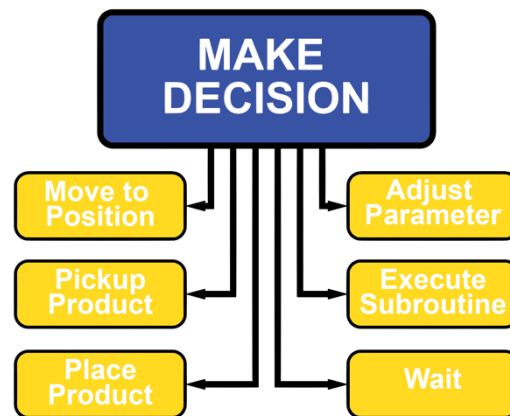


Figure 8 - Generic behavioural Statechart format to discretise robotic processes within manufacturing simulation. Multiple similar states may be defined, and each state corresponds to a chosen action.

The statechart is based around a central *Decision-Making* state. Initially, the state triggers function calls which gather an observation of the environment and pass it to the software agent, the internal logic of which produces a set of Q scores, with the highest-scoring action being selected (Although any policy based on these scores can be used) and returned to the statechart. This triggers the action within the simulated environment, the state moves to that defined by the selected action, and the simulation is advanced accordingly, before returning to the decision-making state. On return to this state, the simulation calculates a reward based on the action just taken, makes another set of function calls to gather an observation, and returns these to the neural network to complete the MDP transition, whereupon it is stored in the software agent’s memory. The network is then trained on the transition, and potentially a randomly sampled batch from its memory, and use of a target network to make forward predictions in a double Deep-Q-Network approach.

5.2. Defining the Action and Observation Spaces

To begin the development of an intelligent agent, consideration should be given to the fundamental aspects of the learning process. These are the state representation, the available actions, along with the rewards provided by the environment. As discussed in the review of the literature, observations may take many forms. They range in structure from low-dimensional array-based representations to full pixel images and consequently require different processing to be interpreted effectively.

The data necessary to sufficiently represent a given system, and to capture the desired knowledge will vary greatly between applications, as such, most intelligent control solutions are bespoke, which requires the definition of a cohesive set of rules for establishing which data sources must be observed to form an effective representation of the state. An effective representation containing the necessary information is a vital first step in developing systems capable of adaptation based on observed changes. Therefore, for the scope of this work, these key capabilities are defined as follows:

- The ability to conceptualise time and temporal patterns, through observation of a time-keeping data source. This enables identification of changes from one event to the next, critical for the analysis of changes over time.
- An awareness of the agents own performance, which is necessary to quantify this value to enable a value comparison with the respective rewards, and how it varies from target performance.
- Conversely, there must be an awareness of other agents' performance, and how this changes as a result of the actions taken by the agent. This may be either directly observed or provided by other modules based on analysis of other factors and to represent various conditions.

-
- Observation of any additional factors that are necessary to model the required dynamics of the system. i.e. anything that may change and influence the system that it should be able to account for.

These abilities allow the intelligent agent to build a sufficiently detailed representation of the manufacturing process, concerning the considered factors relating to the environment and decision-making. Within the context of reducing performance disparity between an agent and its collaborator, the following data points are defined to encompass the necessary information. Combined, these form the *observation space*: [CT, TCT, CT, IT, CIT, SM, BC] for the reinforcement learning agent, defined by:

- The Agent Cycle Time (CT), defined as the time between completion of each action.
- The Target Cycle Time (TCT), given by:
 - Remaining No. Products to Target/Shift duration remaining
- The Collaborator Cycle Time (TCT), multiple of these may be defined for multiple collaborators.
- The Idle Time (IT), which is the consequence of performance disparity and is indicative of the task performance.
- The Collaborator Idle Time (CIT), an additional consequence of performance disparity, this may be further divided into individual observations for each collaborator.
- The Speed Modifier (SM), which represents the current state of the control parameter.
- The Buffer Contents (BC), the number of products in the transition space, necessary to infer the impact of actions on the collaborator.

These observations are combined with those identified in the previous section which are necessary to make predictions about the human collaborators' performance and are additionally used in constructing the state representation used by the reinforcement network;

such as in the case of the elapsed shift duration. These processes are performed by the perception module, which receives the whole observation and divides it into two separate observations to be received by the analytics and cognitive control modules respectively.

An additional consideration that must be made is in terms of normalisation of these observations. This is relevant in the implementation of all machine learning methods and is particularly relevant in developing neural networks as the activation functions of each node can become saturated by large values. This is a challenge in terms of the implementation and training of DQN's, double so as the dataset which will be used for training is not available apriori. To overcome this, a *progressive normalisation* technique is developed and applied, which captures the maximum and minimum values for each observation during the learning process to enable the range of observations to be normalised in the range 0 to 1. This technique compares each new observation to the stored min max values and progressively updates the network with a re-normalised memory each time a new minimum or maximum value is encountered during training. This has a consequence in terms of training time, as the technique leads to initial instability due to initially high frequency of updates, but as the learning space is explored, the dataset stabilises leading to a more effective learning process.

Similarly, to the observation space used to form the state representations of the agent, it's *action space* which formalises the available actions which may be taken by the agent at each step to influence its environment. Similarly, to defining the observation space, the potential number of actions, and how these are discretised and designed is specific to the application and its domain. The combination of these actions provides a range of functionality to perform a wide range of tasks. Consequently, no frameworks for standardising these actions exist, and an action space must be defined within the context of this work, to enable adaptability in behaviours to reduce performance disparity. To achieve this, within the interaction outlines, an agent must able to control and adjust operating

parameters, such as movement speed, force application, etc. to account for differences between its own action and its collaborators and enable adaptability through the use of dynamic control systems. Within the scope of the application considered in this work, the necessary adaptability in behaviour is defined in terms of cycle time, which itself is reflective of movement speed, which is adjusted by controlling a *Speed Modifier*, which acts similarly within the simulation model to the parameters used to affect human task performance. To integrate a reinforcement learning model, which requires discretised action sets, the range of the modifier is discretised into a finite number of steps, with a degree of fidelity appropriate for the application (i.e. smaller steps enable finer control), which will enable the operator to adjust how it performs these operations with respect to its parameters. Consequently, the action space can reduce and simplified to:

[Increase Parameter, No Change, Decrease Parameter]

5.3. Defining Reward Structures

When implementing reinforcement learning through an MDP, the third element of the model which must be defined are the rewards, and how these relate to the selected actions and the environment. This *reward system*, describing how the intelligent agent is rewarded and penalised for different actions and their effects with respects to its goals is again, often application-specific, and an appropriate policy must be established. There must be consideration of the value of these rewards concerning the metrics against which it is being evaluated, but also how these rewards are presented to the agent, in terms of positive or negative feedback, to enables the desired behaviours. Whilst there are numerous examples of Q-Learning, the methods for generating and delivering rewards to intelligent agents are application-specific and must be developed heuristically alongside the agent itself. The policy used for shaping these rewards must consider how these rewards should be presented, based

on which values, to what scale, and at which intervals, all of which will have an effect on training and agent behaviour by altering the environmental feedback it receives.

In a typical reinforcement learning application, agents are provided with fixed rewards which are selected to result in desired behaviours. In this application, the agent may be provided with a fixed reward, R , at each iteration of the sequence, which is then subject to a *Reward Penalty*, which represents performance with respect to the evaluation metrics.

In the scope of this work and the identified interaction model, where the aim is to reduce the disparity in performance, this reward factor may be defined as the proportion of the magnitude of the error between the robotic operators cycle time, and its target performance cycle time, and its cycle time and the cycle time of its collaborator(s).

This reward factor may also be defined so as to represent the impact on other metrics, both positively and negatively. In the case considered, in addition to facilitating the interaction in terms of performance disparity, from the system level, the reduction of idle time is the key aim. As such, the value of observed idle time is also included to penalise the agent for choosing actions which lead to states in which it is observed. The calculation of this reward factor is therefore formalised in Equation 4 and is used to deliver a final reward to the agent, calculated using Equation 5.

$$Reward\ Penalty = (|TCT - CT| + |CCT - CT|) + \sum Idle\ Time \quad (4)$$

$$Reward = R - Reward\ Penalty \quad (5)$$

Furthermore, as each of the values used to calculate the rewards given to the agent represents a different constraint on the agent, different values may be weighting to modify the impact that they have on the reward received. This may be used to *prioritise*, the demands of the collaborator over the production demands, by scaling the rewards accordingly. The inclusion of these weights, w_1 and w_2 , in the reward penalty is illustrated in Equation 6.

$$Reward\ Penalty = (w_1|TCT - CT| + w_2|CCT - CT|) + (w_1IT + w_2CIT) \quad (6)$$

The use of rewards with a fixed value has benefits for the stability of learning, as values of similar magnitude are used (ideally for both positive and negative rewards), avoiding large changes at single steps in learning. Conversely, where changes in the environment are small, the delivered rewards should be scaled proportionately based on how close the agent is towards a balance of its operational targets and amplify the smaller differences in performance when close to the idealised point of operation. The reward factor for the scaled reward policy is calculated using Equation.7 below:

$$Reward = R/Reward\ Penalty \quad (7)$$

This approach may, however, present issues as the reward factor approaches zero, as the reward may grow too large between observations, leading to instability as discussed. This will be evaluated as part of agent development.

As discussed, each parameter update precedes an iteration of the control loop. This is necessary, as the observations used to calculate the reward received are made for the state, s' , and represent the Cycle Times which both the human and robotic operators have just completed and the calculated Target Cycle Time for all products remaining in the shift.

As the rewards are influenced by the effects on the environment, rewards must be delivered in response to actions on either side of an update to the environment, and the choice to take no action should also advance the manufacturing sequence, to avoid a negligible state change and provide meaningful observations, and a meaningful reward.

With respect to the design of reward policy, there are several other aspects which must be defined. A manufacturing scenario is comprised of numerous discrete states and actions which are governed by certain rules. For example, attempting to lower the value of a parameter past its limit is not possible, both in terms of the validity of the action, and the state which it can lead to, as such, there is the inclusion of a fixed negative penalty applied when the agent decides on nonsensical operations, to encourage these actions to be avoided.

Additionally, there are often cases where a non-ideal action must be taken to reach a state where a much more valuable action can then be performed. This is reflected through the use of the discount factor (or gamma parameter in Q-Learning applications), which balances immediate rewards with those that may be obtained from future actions. The influence of the gamma parameter on performance is addressed further in the next section.

Furthermore, consideration of future rewards is also relevant in an application such as this, where the agent aims to optimise its performance against a dynamic variable as it presents additional difficulties when the model is designed as an MDP. The agent must achieve a balance between several dynamic values, which presents issues in terms of learning, as optimisation is performed with respect to a towards a moving target (the future reward, which is predicted based on s' using the same neural network). In the approach pioneered by Deepmind, the use of fixed targets (i.e. a separate network that is periodically updated from the main network) to make these predictions, enables the decision-making neural network to make accurate predictions for expected rewards much faster; and will be leveraged in this approach.

6. Agent Performance Evaluation

This chapter explores the different evaluation methods and techniques used throughout both the development and testing of the intelligent agent and its application. Evaluation is relevant in terms of developing effective neural network learning models, the selection of the relevant hyperparameters, and the process of how these are evaluated. It is also relevant in terms of how the agent's performance is quantified, both in terms of the task and facilitating interaction with its collaborators. Finally, the agent must be evaluated in terms of its efficacy at the system level, through the exploration of multiple scenarios to evaluate the reliability and robustness of the approach. This system-level evaluation is crucial, as it provides insight into the validity of the overall approach, and the extent to which real implementation may be realised. The following subsections outline each of these considerations in more detail.

6.1. Neural Network Hyperparameter Selection

Neural Networks have demonstrated a great deal of success across several domains in recent years, both in terms of prediction and in the application of Q-Learning Agents. The performance of Neural Networks, however, is dependent on multiple factors, including their internal structure, and a variety of hyperparameters, such the learning rate, and the choice of updater function, which describes how errors are propagated through the network. This is an issue when implementing and developing learning algorithms, the chosen value of these hyperparameters is often crucial to the learning efficacy and must be defined in advance. As such, the first stage of evaluation focuses on the performance of these algorithms given different learning parameters to identify potentially viable configurations. This is achieved using a grid search and the previously defined training and test sets. In terms of establishing the predictive accuracy, the method enables an exploration of the hyperparameters and their influence (further contributing to establishing the suitability of the approach), and an optimal

set of values to be identified prior to implementation. The parameter values are iterated over their respective ranges, and neural networks built and trained using their respective test sets. Each network was then evaluated using the relevant isolated test set to produce a *Root-Mean-Squared-Error* (RMSE) score for each parameter configuration. RMSE is calculated using Equation.8, where x is the predicted value, y is the actual value, and n is the number of instances, i :

$$RMSE = \sqrt{(\sum_{i=1}^n (x_i - y_i)^2)/n} \quad (8)$$

Training of the neural networks is performed using a backpropagation approach to iteratively determine the weights for each node. This requires multiple passes through the dataset referred to as *epochs*. A learning rate of 0.01 was selected to mitigate vanishing gradients at the expense of training time; as the output range was relatively small. The Epoch and Number of Hidden Node parameters were evaluated using a grid search approach, with each evaluation performed using the isolated test set. The RMSE was selected as the loss function

The generation of the training datasets is necessary for both online application and the heuristic development of the intelligent agent. To generate the data, simulation runs were performed using a static behaviour for the robotic operators. The data from each of these simulation runs is collated to form a training dataset containing a total of approximately 10,000 data instances, and an isolated test set containing approximately 1000. Concerning training the analytical network, the dataset that is collated consists of data instances formed of the previously identified input features and label. Standard techniques for data pre-processing to improve accuracy were considered, including normalisation, which are explored during the heuristic development of the network.

In the case of the reinforcement learning network, the collated dataset is formed of sets of state, action, reward tuples, which is collected by selecting random actions over several

simulation runs, observing the received rewards, and storing these transitions to memory; 10,000 transitions were again collected. In addition to the training set, a separate test set is defined following the same method, with a ratio between the two of 10:1. These datasets (or memories, more accurately, in the context of RL), are then similarly used to train and evaluate each hyperparameter configuration in terms of how well the network can make predictions; in this application, the expected reward from the action selected given a state. Appropriate normalisation and pre-processing techniques are again applied, following the progressive technique discussed previously.

There are additional parameters which influence the learning of neural networks in a Q-learning approach, to those which define the neural network itself. Within the Q Score, consideration must be given to the future rewards, which are scaled using the parameter gamma, or discount factor, which controls the weighting of predicted future rewards. Another parameter, Epsilon, must be defined and is used to implement Epsilon Annealing. This is used to determine the use of randomly chosen actions, which enable an agent to fully explore the state space. The epsilon parameter is defined using an initial maximum value adjusted with a decay rate for each iteration. The rate of annealing can influence training performance and duration. Both of these parameters should be evaluated where possible, although typical values are $\gamma = 0.7$ and $\epsilon \text{ decay} = 0.99$.

6.2. Evaluation Metrics and Benchmark

Evaluation of the multiple elements proposed in methodology at the system level requires consideration of factors which influence it, with regards to improving the process and the interactions between humans and their robotic counterparts. Most of the constituent elements requiring development which are detailed in the methodology are concerned with learning and are validated in terms of their efficacy as part of their development, using the simulated

environment as a representative task. This is useful in terms of evaluating the performance of the approach in a theoretical sense, however, a method for establishing benchmark performance and enabling comparisons to be drawn must also be defined, to evaluate and validate the benefits of the approach at the system level.

The key criteria concerning the aims of the system have been defined previously but are formalised as the performance disparity here, as minimising the performance disparity is beneficial in terms of facilitating the interactions modelled in this approach. This is for several reasons: Firstly, it reduces the impact of human variation at the process level, both in terms of minimising the development of observed idle-time, and the build-up of product between processes. In addition, a minimal disparity in performance can be said to improve the interactions in terms of fluency, which provides a number of soft benefits from a human perspective and has real benefits for manufacturing systems, by moving closer to an idealised one-piece-flow.

The reduction in disparity must, however, be realised whilst maintaining the demands and constraints of the production process, including appropriate levels of production. This balance between these demands is represented in the reward structures through the target cycle time, which is based on achieving a pre-defined level of production. The key metrics in terms of the system can, therefore, be identified as the cumulative observed idle time of all operators, and the number of products produced. To evaluate the efficacy of the intelligent agents and the proposed approach, a static control case will be defined. This will be done using a separate agent which follows the same operation sequence with a fixed speed modifier. The resultant idle times and productivity can then be used to benchmark the existing dynamics and metrics for the modelled interaction. Evaluation of the agent's performance in terms of the system can then be achieved by replacing the static agent with the intelligent agent, and performing the simulation runs a second time to provide a

comparison. If the approach is valid, the intelligent agent should be able to reduce the observed cumulative idle time, whilst improving on, or maintaining the target level of production.

Establishment of benchmark scenarios is of significant importance in terms of the validation of this and similar approaches, as unlike typical DQN implementations which are based on tasks with performance benchmarks set by human beings (i.e. video games, image classifications, etc.). As a result, the scenario must set a baseline level of performance in terms of the key metrics to compare the adaptable policy against, to determine the efficacy of the intelligent agents, in comparison to existing processes. Furthermore, the benchmarking provides additional validation to the modelling and data parameterisation, by enabling comparison of the modelled values to those observed, to confirm the assumptions and limitations of the modelled interaction, thus ensuring it is suitably representative of the real-world process.

6.3. Evaluation Scenarios

Besides providing a training and evaluation environment for the intelligent agent in terms of learning performance, the simulation model also serves to test the generalisability and validity of the approach, in terms of its efficacy at the system level, in terms of the metrics outline previously. To validate the efficacy of the approach, different simulation environments will be explored to measure the extent to which the agent can achieve consistent performance over multiple cases. This is achieved partially by adjusting the simulation parameters as discussed to represent different conditions and interacting behavioural changes in the operators and will partially be addressed by varying the task and production demands. Furthermore, the simulation environment explored in this work is based on, and representative of, an existing real-world problem and process; which, by comparison

to, and demonstrated improvement over, a static control case, will lend additional validity to the approach.

The simulation model is already designed to provide a wide range of different scenarios in terms of the collaborators' demands. The parameters which are used to govern human behaviour, as discussed previously, can be modified to provide a variety of behaviours. This provides a good initial variety of cases in which to evaluate the intelligent agent's performance in terms of its learning ability, and robustness in terms of delivering the system-level demands. These scenarios will be further explored by varying the demands of the process in addition to those of the collaborator. This will be achieved by using different production targets to put more strain on the robotic agent, to see if it can optimise in this scenario and whether it may leverage an even greater level of performance.

In addition to considering the demands on the agent, it is also necessary to evaluate the agent's performance in terms of how robust the agent is when these demands have different effects. To explore this, the simulation model will be modified to compare the performance of the agent when in both the upstream and downstream positions of the interaction, to see if a consistent performance can be achieved in both cases. This concept will be further expanded by considering multiple collaborators and how the demands of both an upstream and downstream collaborator can be balanced, if at all, by the learning agent. This will be in terms of both multiple robotic operators (each with their own intelligent agent), which extends and reconfirms the previous study in terms of a larger system; and multiple human collaborators, to test how resilient and capable the proposed methodology is to reduce the total impact of disparity at the system level. These configurations are illustrated in Figure.9. Furthermore, the use of multiple agents provides a good evaluation for the applicability of the work (Young *et al.*, 2011) suggesting that embodiment of these agents enables different behaviours to be observed based on individual cumulative experience.

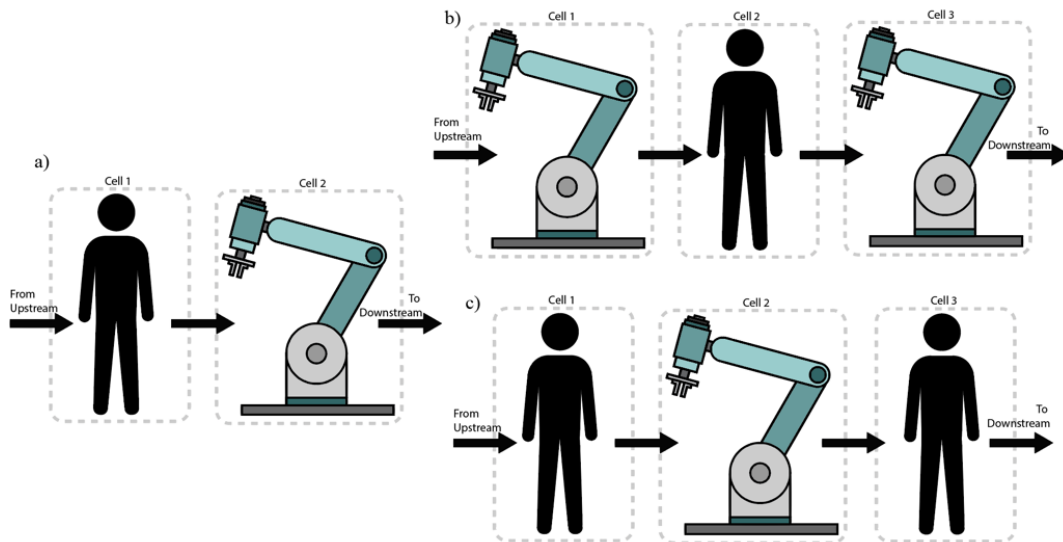


Figure 9 - Visualisations of the configurations of different operators to be explored to ensure the robustness of the approach.

6.3.1. On-Task Validation

Finally, the exploration of performance in these different configurations will enable a robust agent to be evaluated through the use of a comprehensive model of a case-study process, to be introduced in the next section, which is representative of a real-world process. Evaluation in the form of a case-study is particularly important, as it provides a representative problem, informed by verifiable data, by which to validate the task performance of the agent, and the potential benefits and implications.

This is for the dual purpose of evaluating and demonstrating the simulation and learning methodologies and providing an exploration of how they can be practically applied to a manufacturing process, and how this application can provide the theoretical benefits identified by the literature. With respect to the first point, the use of a real-world process enables the methodology and intelligent agents to be evaluated against a *ground truth*, in terms of data collected from reality. It enables the effects of behavioural variations at the

process level to be validated against similar metrics monitored in the real-world process, providing additional and crucial validation for an otherwise abstract approach.

The case-study additionally provides a significant contribution to the work, by demonstrating the application of the developed methods to a process, and how a real-world scenario can be decomposing into the relevant abstract elements to enable the use of intelligent agents. The practical application of the method is also crucial to evaluate, as the research gaps in the area remain issues surrounding realising the benefits that these emerging technologies promise to deliver. With respect to the latter point, the adaptability that decentralisation and reinforcement learning provides is hypothesised to enable the intelligent agent to adjust its operating parameters to affect behaviour that reduces the performance disparity between itself and its collaborators. This behaviour is further hypothesised to improve these processes at the system level, by converging to an optimal policy which reduces the observed idle time and WIP levels within the process; improving the overall productivity and leanness of the process. In addition, the case-study provides further ground truth for human performance, and validation of the modelling approach. This environment will explore various parameterisations of human behaviour, including the use of human-derived parameters, to provide an evaluation as to the capabilities and potential benefits and applicability of the work to real-world implementation.

Altogether, the case-study is a necessary process for the evaluation and validation of the developed methods, both in terms of establishing the efficacy of the approach, and how the approach demonstrates many of the desired behaviours and benefits which intelligent manufacturing promises. Realising the constituent elements into a cohesive system capable of delivering results is a key aim of this work, identified as a part of the initial research questions, and consequently, evaluation in such a manner is critical in order to draw conclusions about this work as a whole.

7. Case Study

This section presents the application of the research framework and proposed methodology to a case-study, conducted in collaboration with an industrial partner, Olympus Surgical Technologies Europe (OSTE). Based in Hamburg, Germany, and reporting back to HQ in Japan, the Cardiff site of OSTE forms part of a multinational design, development, and manufacturing process for the production of high-quality, advanced medical equipment. The Cardiff site is responsible for the inhouse design, testing and manufacture of multiple single-use surgical devices, with multiple fabrication facilities and production lines.

As the products in this case are medical devices, the manufacturing facilities are enclosed in a clean room, and there is a great deal of benefit to automating as much of the manufacturing process as possible, to prevent biological contamination. However, the delicate nature of certain stages of the assembly process require human dexterity and adaptability. As a consequence, the automated production line at OSTE provides an excellent case-study opportunity, and enables the generalised approach developed to be applied to a representative process, with the potential to deliver tangible and actionable results. The production process is visualised in Figure.10 below.

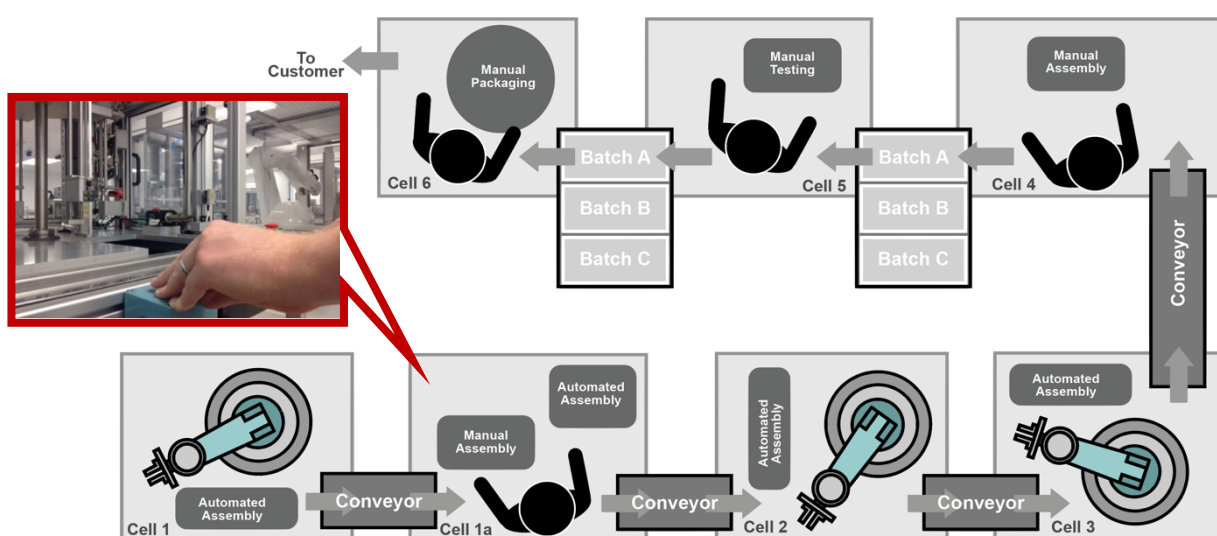


Figure 10 - Diagram of semi-automated production line used for case study. Call-out detail illustrates manual operation cell.

This case-study highlights the issues associated with a critical human presence within such systems. From Figure.10, the operation at *Cell1a*, represents the position of the human element within the overall process. The particular assembly operations at this stage involve the delicate manual insertion of a fine copper electrode wire to the product, some semi-automated fixing operations, and a visual confirmation check that must be performed.

The remaining operations in the process are fully automated, and run on a repeatable sequence of operations triggered by PLC's. As such, the automated elements have a much smaller variance in their cycle times for each product, and the system is optimised to a nominal operation speed. The consequence therefore, of the human operator, is the significantly increased variability, both temporally and between individuals, in performing the assembly operations at this position. This is also exacerbated in this case, as the manual operation exists between two automated operations, doubling the potential impact of a variation from the nominally defined cycle time for the whole process. As such, the key challenges presented for the intelligent agent, are the ability for the upstream and downstream operators to account for the variation in the manual operation, and deliver/receive products from the human cell with as minimal a delay as possible. This will alleviate the observed idle times and prevent the build-up of in-progress pieces in the interstitial buffer zones, whilst maintaining a level of productivity consistent with the most variable element of the process.

7.1. Manufacturing Process Modelling

Following the presented methodology, a case-study scenario was developed based on the previously detailed real-world semi-automated production line. A model of this process was created using the simulation platform to both provide a test environment for the developed software agent capabilities, and to demonstrate the feasibility and benefits of the approach, to these processes as a whole. The real-world production line is responsible for the manufacture

of single-use medical suction devices and includes several automated operations grouped into individual cells and a manual operation which occupies its own cell. As such, this interaction between the first and second stages of the process provides an ideal case study, as it can be suitably generalised as identified in the methodology. These two cells constitute the start of this manufacturing process, and the interaction as it is applied to the case study is illustrated in Figure.11.

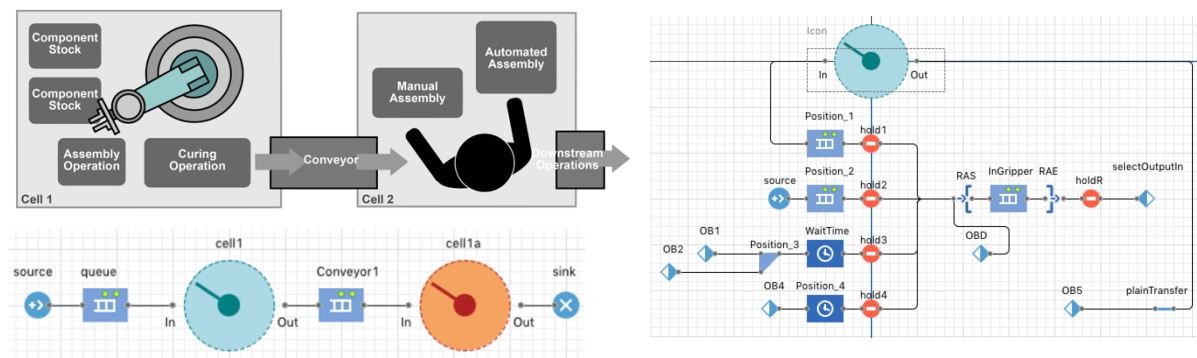


Figure 11 - Case Study production line, illustrating the interaction between the robotic cell and human-operated cell, and the simulation model for the robotic operator developed in Anylogic.

The first manufacturing cell (cell 1), consists of a robotic operator which is responsible for the completion of several operations. These are the retrieval from stock and assembly of two components, with a gluing and curing operation. Once these are complete, the operator is also responsible for passing the completed product to the intermediary conveyor where it is moved to the second cell, which contains the manual assembly step. This manual step involves the careful insertion of a fine electrode wire and is a good example of both physical and cognitive loading; involving both dextrous manipulation and visual analysis. The conveyor has a discrete capacity of 10 products, which, when either empty or full, leads to observable idle time in each operator. This case provides a good representative real-world example of the modelled interaction, and the conditions relevant to the human factors which influence task performance.

The manufacturing operations performed by cell 1 is discretised according to their type based on the presented methodology. The retrieval operations are represented by *queue* elements which hold a specified number of products, whereas the assembly and curing operations are represented in the simulation environment by *delay* elements given their associated fixed duration, which hold a product for a specified time. The operations are represented by the elements down the left side and the block to the right acts as the robot's manipulator in Figure.10. The gripper element is then able to distribute products around the simulation environment using the selective output blocks and an associated position, with delays for moving between them. The actions which therefore enable the robotic operator to complete these operations are again discretised based on the methodology and are defined for this case as the ability to: move to each position, pickup and put down a product, wait for a defined period, and manipulate the relevant process parameters. These actions are combined to form the statechart illustrated in Figure.12.

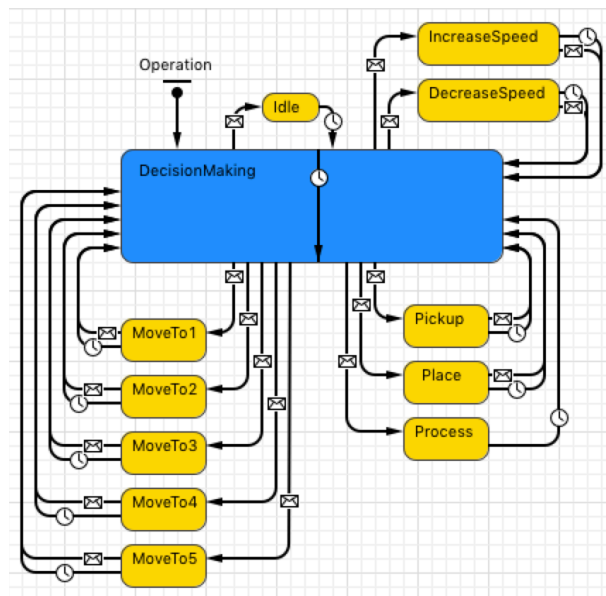


Figure 12 - Statechart used within Anylogic to determine robotic operator behaviour, each yellow state is an action

The observation space of the agent is also defined for the application following the methodology identified. From the simulation environment, the software agent receives a state observation in the form of a 1D array values representing the data points or input values. These values, as discussed, will vary between applications, but take a generic form, and in the presented case study are defined using 6 variables: the Robotic Operator's previous Cycle Time (CT); The calculated Target Cycle Time to reach the production target based on the remaining shift duration (TCT); The cycle time(s) of the human collaborator, the Collaborative Cycle Time (CCT); the current parameter modifier, the Speed Factor (SF); and the Idle Time (IT) for the last cycle:

[CT, TCT, CCT, SF, IT, CIT]

The action space of the agent is again defined using the methodology, and in this case, forms an array of actions enabling the agent to both increase and decrease the value of the speed modifier parameter, effecting the duration that actions take within the environment, or complete the next task without adjusting the parameters:

[Increase Speed, No Change, Decrease Speed]

Additionally, timings were made of the second cell in the real-world process, and the recorded values averaged to develop nominal cycle times for each operator, to enable representative modelling of the human performance and the interaction as a whole. These parameters for human performance were defined as in Table.3. based on a combination of observations and work examined in the existing literature.

Table 3 - Values used to profile performance for each of the three operators based on observed case-study

Operator Number	1	2	3
Base Cycle Time (s)	40	45	50
Fatigue Modifier (End of shift)	1.2	1	1.1
Shift Modifier AM	1.1	1	1.1
Shift Modifier Middyay	1	1	1
Shift Modifier PM	0.9	1	0.9
Weekday Modifier Monday	1		1.1
Weekday Modifier Tuesday	1		1.05
Weekday Modifier Wednesday	1		1
Weekday Modifier Thursday	1		0.95
Weekday Modifier Friday	1		1

7.2. Human Performance Profiling

A set of task observations was made of a set of sequential, manual assembly processes to illustrate how to apply an initial monitoring period to human operators to capture an effective representation of their task performance. This was done using a different production-line assembly process similar to many automated layouts, which is entirely manual. The operations of this process closely resemble those performed in the semi-automated process identified previously, consisting of delicate, manual assembly of small components, completed with visual checks at each stage. As such, the variations in performance due to the type of task are likely to be consistent with those observed for operators in the automated process, and the work schedules for each location are also similar, preserving the effect of other fatigue influencing human factors. This was done, as each of the individual processes remains easily delineated, facilitating monitoring in terms of the cycle time at each position.

The production line was designed to operate as close to a condition of One-Piece-Flow (OPF) as possible and provides a source for data with a 5 individual observation points improving the total observation frequency, by allowing several different individual profiles of human performance to be collected simultaneously.

Cycle Time measurements were performed for each of the process positions, to build a dataset of observations of task performance for each operator. Five measurements were taken at each interval with each interval spaced two hours apart. This process was repeated across two days, and across two work shifts, increasing the number of observations used in the profiling, and building a dataset to verify the assumptions regarding the impact of human factors. These values are then scaled to match the nominal time of the case study application. Visualisation of the raw dataset by way of a heatmap aids in understanding the distribution, as does plotting the data values such as the mean, variance and standard deviation values for each operator, over all conditions. These are illustrated in Figure.13.

Operator	PM				AM			
	1	2	3	4	1	2	3	4
W E D N E S D A Y	49.08	35.25	49.77	46.20	38.79	44.76	31.26	48.06
	42.30	35.43	48.84	47.70	38.37	41.64	25.80	49.89
	45.54	39.21	46.41	47.22	39.30	47.70	34.80	48.03
	43.62	39.00	56.76	45.66	49.80	49.17	27.54	49.77
	40.26	41.91	49.74	44.52	38.73	40.47	26.28	45.36
	37.68	56.34	40.98	44.22	40.86	44.55	56.67	40.41
	33.63	46.35	52.44	46.50	40.71	41.01	63.45	45.57
	34.17	50.07	45.51	63.42	38.91	43.65	45.09	40.44
	33.15	40.11	56.31	44.25	57.99	43.38	49.35	41.46
	34.29	39.66	53.82	43.32	47.04	44.76	51.81	47.22
T H U R S D A Y	41.70	35.76	43.50	56.22	45.36	45.39	41.91	53.79
	34.65	37.14	55.11	47.28	41.91	45.90	54.51	46.14
	41.01	48.51	41.07	46.62	45.09	44.34	32.70	48.57
	39.27	32.37	46.41	45.90	41.76	43.71	36.15	41.31
	51.93	39.51	52.44	48.21	52.17	41.40	38.70	41.19
	36.57	42.66	48.78	42.15	36.15	67.89	44.16	42.06
	38.22	44.22	37.11	43.68	47.25	39.30	47.43	38.61
	42.30	28.32	45.75	46.23	41.37	35.01	45.60	42.72
	42.18	35.61	47.97	47.79	49.80	31.83	42.45	43.50
	39.03	38.67	47.85	49.47	41.10	41.10	37.32	48.00
Mean	40.03	40.31	48.33	47.33	43.62	43.85	41.65	45.11
Std dv	5.09	6.47	5.27	4.79	5.58	6.93	10.45	4.04

Figure 13 - Heatmap illustrating the distribution of cycle time, mean, and standard deviation for all operators, across all conditions.

Furthermore, it is necessary to validate the assumptions made in the methodology regarding the appropriate probability distribution to describe the variation observed for a given task. This is achieved, by visualising the collected data as a histogram to confirm that the assumptions of its form are valid, and to identify appropriate parameters which must be defined. The histogram produced for the observed dataset is illustrated in Figure.14.

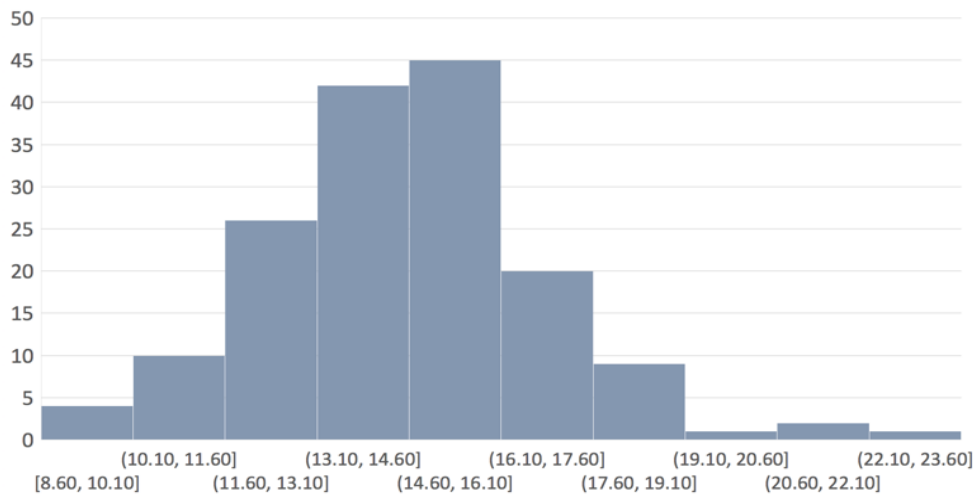


Figure 14 - Histogram showing the distribution of cycle times for all operators across all conditions.

The distribution of observed cycle time for the process can be seen to best approximate a normal distribution centred around a mean. As such, each human performance profile will use a Gaussian distribution which is parameterised by the respective derived mean and standard deviation.

As discussed in the methodology, the next step is to derive parameters from the dataset, from the average task performance. Although these parameters may be derived along any of the variation vectors identified for the dataset, to enable the influence of different contexts to be compared, to support assumptions made in the use of the modifiers and results of related work, or potentially directly used as variable parameters. However, in this application, the monitoring process lacked the appropriate fidelity to generate individual parameterisations for a suitable number of scenarios. The dataset contains three key vectors: Time of day AM vs PM; Day of the Week (Wed vs Thurs); and the Individual Operator. The dataset is

aggregated into individual subsets for each vector from which mean value and standard deviation are calculated; the resulting parameters derived are illustrated in Figure.15. Additionally, the final column shows the calculated average values used to replace the nominal times previously used. The selected operator profiles used are shown in Table.4.

		Variation Vector/Combination of				
Operator No.		Wednesday Samples	Thursday Samples	All AM Samples	All PM Samples	All Task Samples
1	Mean	41.21	42.44	43.62	40.03	41.83
	Std Dev	6.02	4.90	5.58	5.09	5.45
2	Mean	43.22	40.93	43.85	40.31	42.08
	Std Dev	5.04	8.12	6.93	6.47	6.76
3	Mean	45.63	44.35	41.65	48.33	44.99
	Std Dev	10.57	6.16	10.45	5.27	8.56
4	Mean	46.46	45.97	45.11	47.33	46.22
	Std Dev	5.51	3.00	4.04	4.79	4.51

Figure 15 - Derived parameters for each of the variation vectors considered. Means and Std. Deviations used to parameterise a Gaussian distribution.

Table 4 - Values used to parameterise the performance distribution for each of the three operators based on observed case-study

Operator Number	1	2	3
Mean, μ (s)	41.8	42.1	46.2
Std. Deviation, σ (s)	5.5	6.8	4.5

7.3. Agent Development and Training

This section provides a detailed overview of applying the training methodologies outlined in the previous chapter to the outlined case-study. As has been discussed at length already, machine learning applications require the definition of various hyperparameters which must be known apriori. These parameters vary depending on the application and consequently, there exists a degree of heuristic development that must take place to establish effective learning models. The section outlines the process of identification for both the analytical and

similar decision-making networks, but with distinct differences in their methodologies, as discussed in the previous section. Both of these networks were constructed using the ADAM updater, with rectilinear (RELU) activation functions for the hidden layers and an IDENTITY output layer, to enable real values to be produced by the networks.

7.3.1. Human Task Performance Prediction

The neural network used for the analysis of the human factors was trained as discussed in the methodology section over several different parameter conditions, and these results recorded and visualised to identify the optimal hyperparameter combination, in the presented case, the network structure, in terms of both the number of hidden layers and the nodes in each layer. The simulated process and the agent were defined as previously discussed and used to generate both training and test datasets for use of back propagation, which requires multiple passes through the dataset referred to as epochs. Too many training iterations may lead to overfitting, but it unlikely for this dataset as the feature set is relatively small compared to the number of data instances.

The network also requires the specification of a learning rate, of an appropriate value to enable efficient training. Within the presented case, there were problems early in development, as the range of the output variable is relatively small, compared to the magnitude of change in observed values. Consequently, the gradients would often converge and become stuck at local minima. The inclusion of dropout to the hidden layers, using a rate of 1, to randomly remove a connection at each training step also reduced this effect and improved the predictive performance. Consequently, a rate of 0.01 was selected to mitigate vanishing gradients at the expense of training time. The relevant number of iterations, the number of hidden layers, and the Number of Hidden Node parameters were then evaluated using the grid search approach as outlined in the methodology, and the results are presented in Figure 16.

Presentation of these results in the form of a heatmap helps to visualise the variations in the performance of the network in different conditions. This enables several initial conclusions to be drawn, concerning the selection of the optimal combination of hyperparameters. From the heatmap, it can be seen that the majority of trained networks converge within a small number of epochs. In most applications initial improvement is rapid, as initially randomised or estimated weights and continuous outputs are quickly reduced to

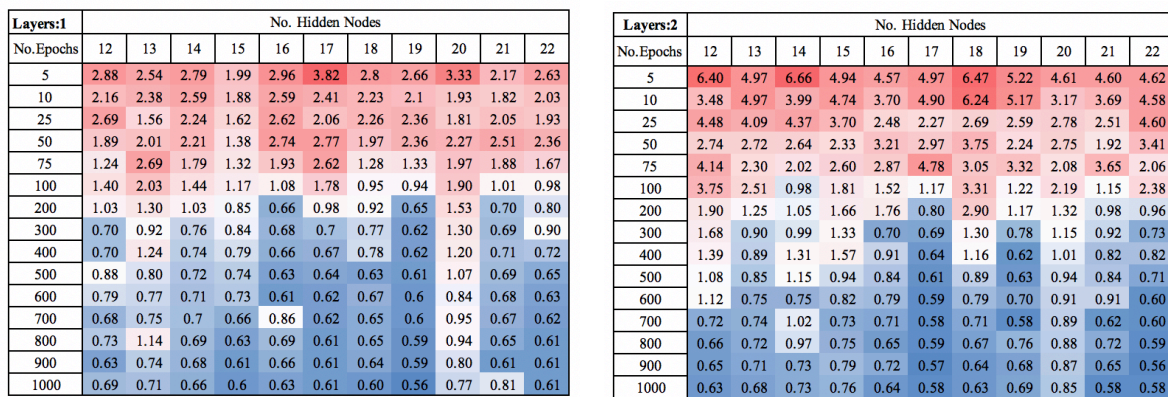


Figure 16 - Heatmap of RMSE scores, for the number of hidden nodes against the number of epochs trained. For a) 1 hidden layer and b) two hidden layers.

the observed range, minimising the error significantly. However, as the output range of the target value (the predicted cycle time) is relatively small, a subsequently longer training period is required to reduce the error to a level which enables differentiation between instances.

Three potential network configurations can be identified from the heatmap: the 15-hidden-node, 18 hidden-node, and the 22 hidden-node, single-hidden-layer configurations all demonstrate comparatively low RMSE scores after 500 training epochs and are potentially viable. This also suggests that training in addition to 500 operations leads to diminishing returns in accuracy with respect to execution duration. The additional hidden layer can be seen to introduce higher initial errors and slower, more unstable convergence, for generally small gains in predictive accuracy. Of the identified configurations, the 15-node

configuration has a low initial error, the most stable rate of improvement highlighted by the gradual change as the training progresses. In addition to this, the simpler configuration reduces complexity and computational demands, additionally benefiting from faster training times. As such, a single-layer, 15 hidden-node configuration was selected for use in the simulation, despite lower absolute RMSE scores in the 18 and 22 node configurations.

7.3.2. Reinforcement Learning

As with the analytical network, the decision-making network was also evaluated using the grid search approach detailed in the methodology to establish the optimal hyperparameter combination. Figure.17 presents the grid search results for the DQN; in the case of training the reinforcement agent, each observation is guaranteed to be observed only once (although possibly multiple times due to the use of experience replay), and consequently, only a single training pass is performed for each combination of parameters.

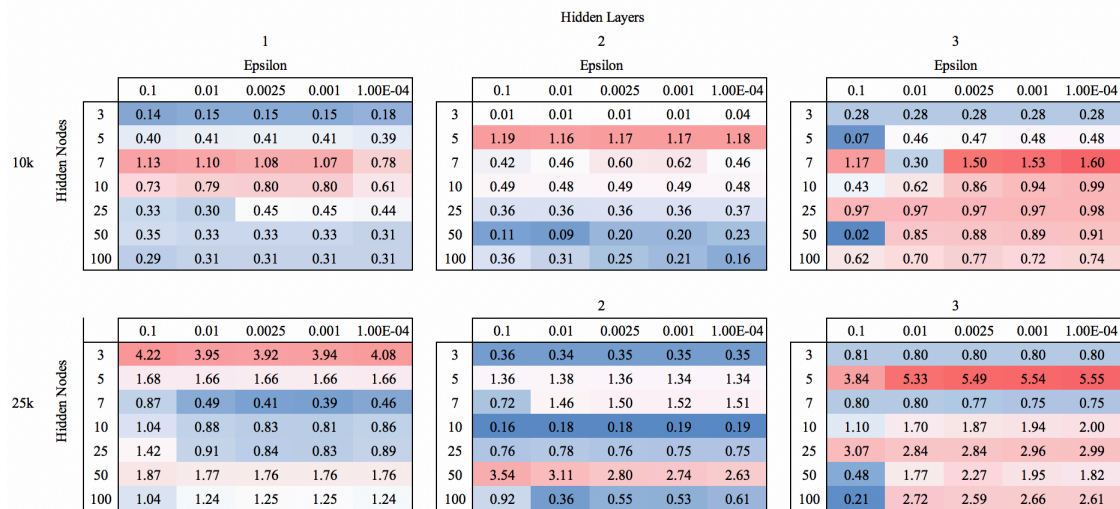


Figure 17 - Hyperparameter grid search results for DQN Agent using a)10k training instances, and b) 25k training instances. RMSE scores represented by the colour (blue lowest).

Furthermore, to evaluate the efficiency in terms of the capacity of the intelligent agent's memory, two datasets were defined containing 10,000 and 25,000 instances. In the

case of Figure.17, anomalous results (whereas in the majority of cases, the network achieved a reasonable level of accuracy, for certain hyperparameter combinations, accuracy was exceptionally poor, skewing the range of results heavily) were excluded from the colour-mapping, to provide a clearer indication of performance variation.

The optimal combination of hyperparameters in terms of predictive accuracy of the network can again be identified from the heatmap as the result with the lowest overall RMSE. The deepest blue values in Figure.17 suggest that the two hidden layers and 10 hidden nodes configuration achieves the lowest and most consistent error for multiple learning rates. The 3-Hidden-Node, 2-Hidden-Layer configuration is also plausible. However, based on the results of the analytical network parameter selection, and the heat map results, there is evidence that simple structures provide a generally better choice for several reasons. The single hidden layer 3-node network similarly achieves good accuracy, although too few hidden nodes may contribute to over-fitting of the model.

Consequently, a 5-hidden-node network as also selected to evaluate a simple model's performance, at the benefit of faster training, as again, diminishing returns exist with increased complexity. The learning rate can be seen to have a real, albeit negligible influence on the performance of the network, with the networks internal structure being a much better dictator of predictive ability. As such, a Learning rate of 0.0025 will be used moving forward, to achieve a balance in overall performance. In terms of the effect of training set size, and how this may reflect on the number of transitions stored in the intelligent agent's memory, the results of the preliminary work suggest that there is no clear advantage to larger sample sizes. The 25,000-instance dataset does indicate an overall better predictive ability but also produces some of the worst-performing combinations, likely as a result of the larger potential range and variance in the samples. As such, a working memory size of 10,000 will be used for the intelligent agent in practice.

The manufacturing process identified for use in the case study also continues into a second automated cell following the manual operation. Consequently, this enables the model to be expanded, and the capability of the approach and the intelligent agent evaluated using a different generalised environment and interaction. Furthermore, the case-study model becomes more representative of the real-world process, lending further validity to the approach and its application in a real-world process, and consequent tangible benefits. In the downstream position, the agent has far less influence over the process as a whole and is entirely dependent on the pace set by the upstream human collaborator. This downstream process is illustrated similarly to the upstream and manual process in Figure.18 and is again discretised into its relevant operations and the elements which represent them in the simulation environment. The observation and action spaces for the agent remain the same as in the upstream case.

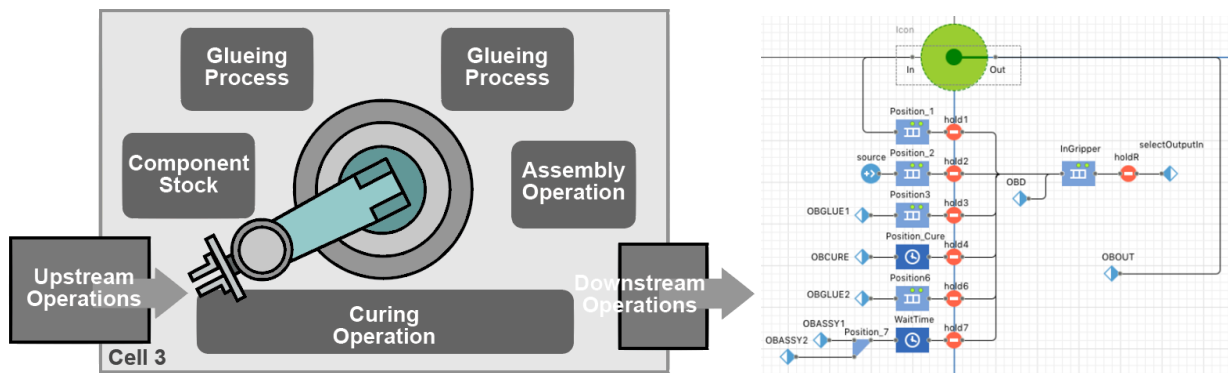


Figure 18 - Illustration of the downstream automated process in the case-study production line, and the corresponding simulation model.

The grid search approach is repeated for the downstream agent, the scores illustrated in Figure.19, and the optimal hyperparameters are chosen as before. Once more, the heatmap enables optimal hyperparameters to be established. The lowest scores were attained by a 25 Hidden-Node, 2 Hidden-Layer, and 10 Hidden-Node, 1 Hidden-Layer configuration. To test the applicability of the potential for embodiment to enable effective behaviours given different experiences and beliefs, the previous configurations for the upstream agent were

evaluated alongside. Convergence of these networks under the different interaction dynamic would lend weight to this theory.

The case-study provides a representative, yet suitably generalised environment to provide both heuristic development of the intelligent agents, and an evaluation platform with regards to the benefits of the approach the system level. The following section present the results of the development process for the agent implementation, and the evaluation of the agents, over a variety of situational constraints.

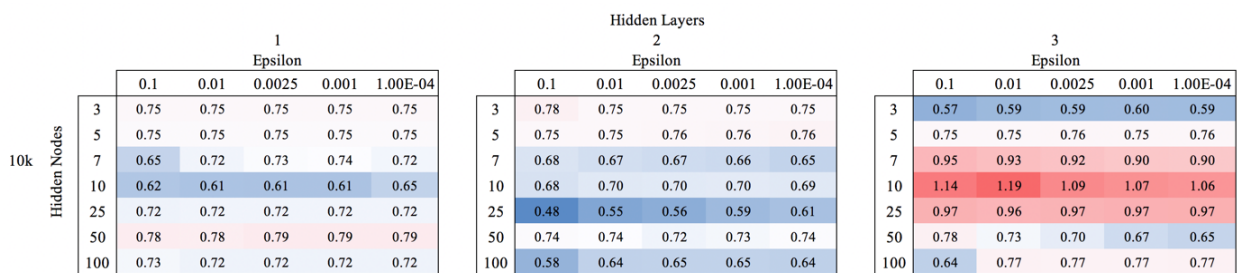


Figure 19 - Hyperparameter grid search results for downstream DQN Agent using a)10k training instances. RMSE scores represented by the colour (blue lowest).

8. Results

This section presents the heuristic aspects of the development and application of the methodology and the intelligent agent to the identified real-world manufacturing interaction.

This section is divided into three sub-sections, the first outlines the results of human-performance prediction, the second, which covers the development and application of the reinforcement learning methodology to the real-world case, and the third, which evaluates performance at the system level, with respect to the identified metrics.

8.1. Task Performance Prediction

Once the optimal parameters were established, the simulation was again run using the trained network, with a random seed to produce a previously unseen environment. This was done for a full three weeks of operation, to evaluate predictive performance over varied conditions.

Random element seeds are changed between runs, ensuring appropriate variability when the same conditions are imposed. Figure.20 plots the simulated human operator cycle times against the analytical neural network's predictions for simulation runs over a working week.

Comparing the predicted values to those observed enables the performance of the neural network to be evaluated in terms of how well it can identify the influence of human factors and variation between operators. Figure.21 again plots the observed cycle times against the predictions made by the analytical network but illustrates the same day for different shift orders to illustrate how predictive ability varies with respect to this parameter.

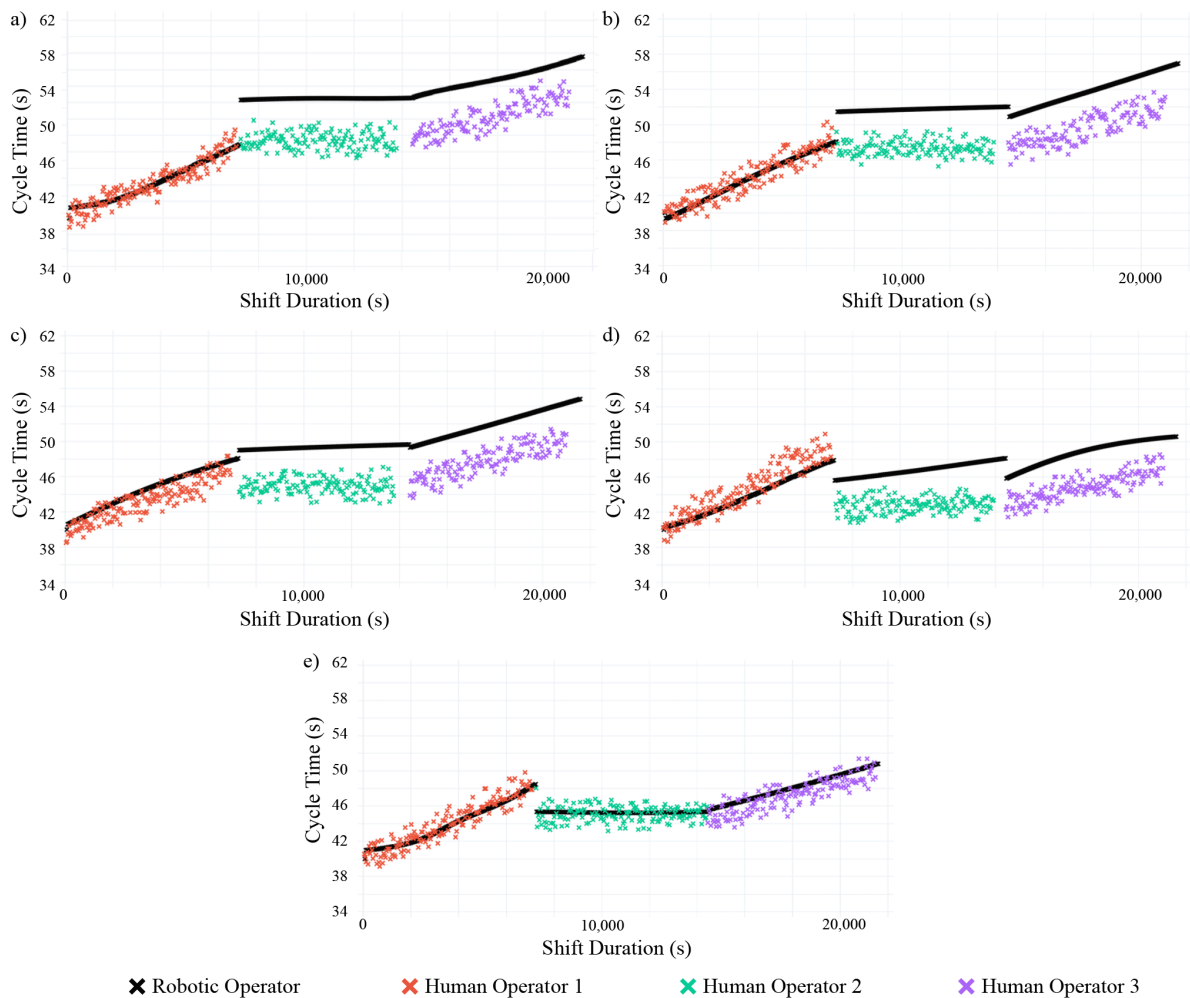


Figure 20 - RO CT and HO CT over time for a) Monday, b) Tuesday, c) Wednesday, d) Thursday, e) Friday.

From the presented figures, some initial conclusions can be drawn regarding predictive performance. The cycle times predicted by the analytical network show a clear trend that closely follows the observed human cycle times, a pattern which is consistent over each of the scenarios. This suggests that the network is able to successfully identify the disparity in task performance which exists between each of the individual operators. Furthermore, the predictions of the network can be seen to increase with the duration of each human collaborators shift proportionately to the observed decrease in task performance. This is indicative that the network can predict the effect of time-on-task fatigue.

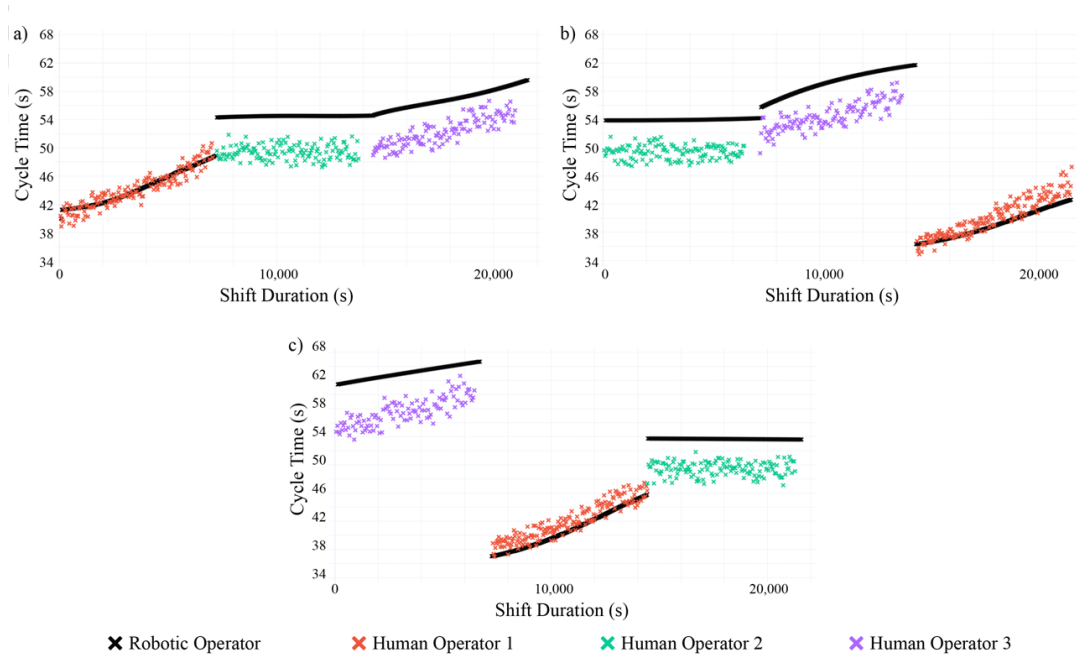


Figure 21 - RO CT and HO CT over time for Monday shift of three weeks and corresponding shift orders a) 123, b) 231, and c) 312.

Within Figure. 20 a) through e) present the network predictions for each day of the first week, maintaining a constant shift order and consequently the impact it has on human task performance. The predictions made by the network can be seen to decrease with each weekday, in proportion to the influence of the weekday modifier, suggesting the network can distinguish successfully this influence. Additionally, Figure.21 compares the same weekday for each of the shift orders. The predictions made by the network suggest that can account for the influence of the time of day, as the predictions can be seen to move consistently when comparing am and pm shift performance of Operators 1 and 3, but not in the case of Operator 2 who is not susceptible to the influence of the shift modifier.

The offset from the observed times in the results also suggests, however, that the output range for each of the individual operators having significant overlap, and the contradictory influences of the multiple factors. The observed ranges of Operators 2 and 3 are often similar, whereas Operator 1 more consistently occupies an output range with limited overlap. Despite this, Cognitive Control these results provide good initial evidence that a neural network-

based approach has an application in predicting, and consequently accounting for, variations in human task performance.

8.2. Reinforcement Learning for Cognitive Control

This section explores the final heuristic development, the training, and initial performance evaluation of the intelligent agent in terms of determining appropriate action policy. The training was performed as discussed in the methodology, and the simulation environment parameterised for different human operators. Two agents were trained using the 5 hidden-nodes and a single hidden layer, and 10 hidden-nodes, and two hidden-layers structures, previously identified from Figure.17 as being optimal for this application. The agent's scores were recorded for each training iteration and are illustrated in Figure.22.

Training episodes represents a full day of real-time operation, and the parameters are iterated to enable the agent to explore as much of the state space as possible. Epsilon annealing was utilised as discussed in the methodology, with a rate of 0.995.

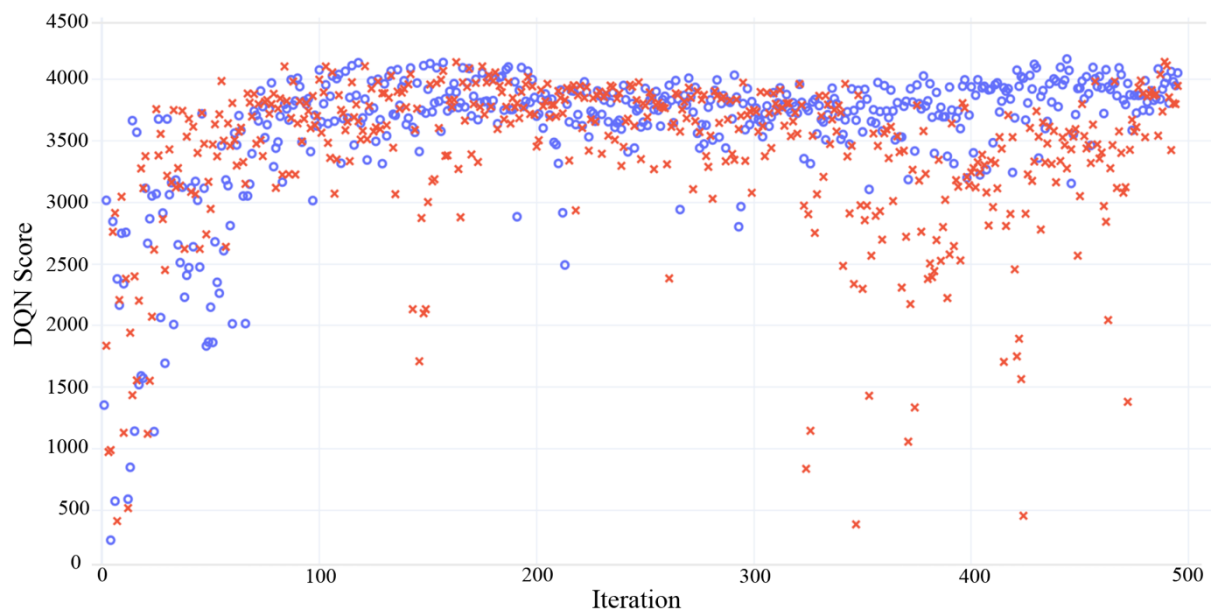


Figure 22 - Results of DQN Agent training over 500 iterations for both the 5 Hidden Node and 10 Hidden node configurations.

The agents are then evaluated over another full simulation run of 15 iterations (with further training suspended), to evaluate performance in terms of key metrics at the system level. The values of observed idle time, the cumulative score received by the agent, and the productivity are presented alongside those for the static case (these values are also recorded for the Static case and used as a benchmark for intelligent agent performance) in Figure.23. The static case is used as a benchmark, as this policy best highlights the impact of human variation on the process, with different system-level implications dependent on multiple human factors. Figure.22 shows that both of the tested configurations converge to a solution, whilst Figure.23 illustrates that both generate an effective action policy after training leading to decreased observed idle time and maintained levels of production; when compared to the results of the static case. This can be seen below as consistently lower observed idle times when compared to the static case, with minimal variation to overall productivity, across all evaluated conditions.

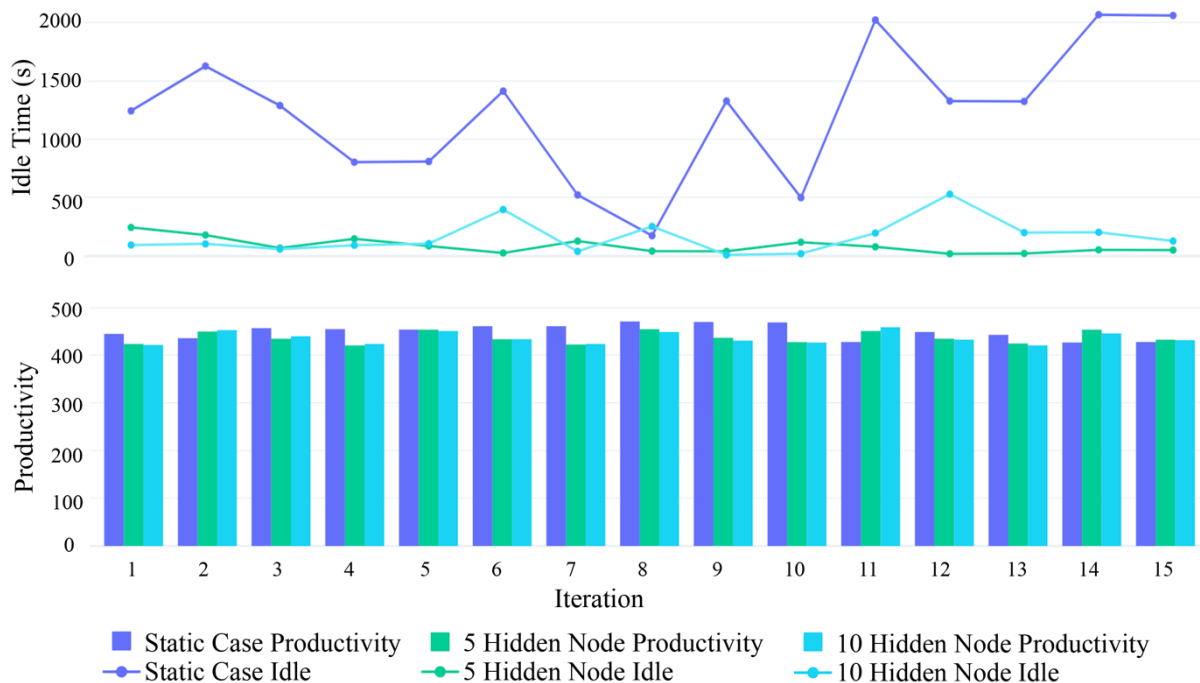


Figure 23 - Comparison of DQN Agents to benchmark case in terms of Idle Time, DQN score and Product Count.

8.2.1. Reward Structures

After evaluating the hyperparameters, and developing an effective set to enable learning, the influence of how rewards are presented to the agent must be considered. This is in terms of both the reward structure and effect of discounting future rewards through the gamma parameter. The same methodology as before is repeated, with the single-layer 5 hidden node configuration selected for its stability and decreased in training time. This was done for each of the reward structures outlined previously. For each case, a value of 10 is used for R, representing the reward for completion of a task. Invalid actions receive a penalty of -5. The evaluations of both the penalty (Equation (5)) and the scaled reward structure (7) are repeated for gamma values of 0.1, 0.7, and 0.9. To explore the influence on both learning and policy generation. Figure.24 illustrates the training of these agents' structures in terms of the score achieved by the agent.

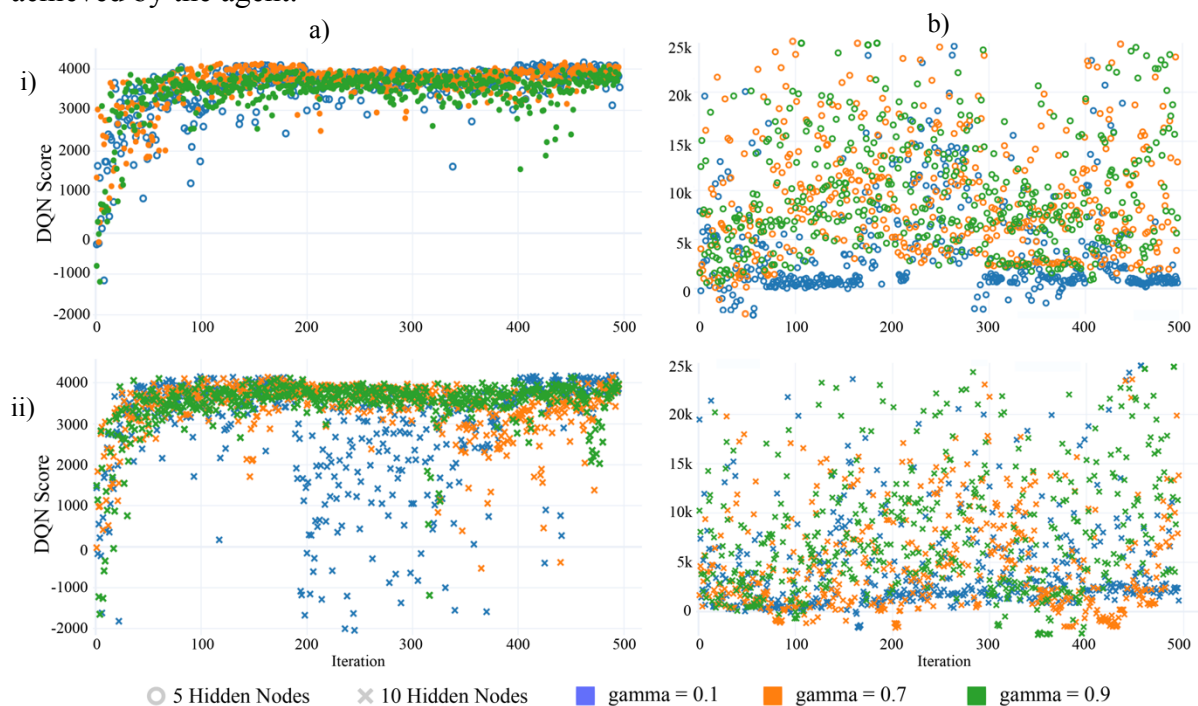


Figure 24 - Comparison of DQN structures i) 5 Hidden Nodes ii) 10 Hidden Nodes, and training in terms of DQN score for different gamma values and reward structures: a) Penalty rewards, and b) Scaled rewards.

The results presented in Figure.24. show that the scaled reward policy can be seen to be much more sensitive to the gamma value, with none of the evaluated networks showing any indication of convergence to a consistent policy. The penalty-based structure can conclusively be considered the more effective policy for generating reward for agents in this environment. Furthermore, it can be seen that the gamma parameter has a negligible effect on learning or training performance, for either of the evaluated policy, indicating that the stability of rewards is important to enable effective training. Consequently, this policy along with the single-Layer, 5 Hidden Node configuration as a benchmark, and a gamma value of 0.7 to balance current and future rewards, will be used for evaluation and training moving forward.

8.2.2. Parameter Variation

With a suitable reward structure now established, the implementation can be considered fully developed, and the agent is more intensively evaluated in terms of absolute performance and generalisability, by testing over multiple scenarios, as discussed in the methodology. The simulation is parameterised to represent the influence of multiple human factors which affect fatigue and introduce performance variation, and the intelligent agent is evaluated across different scenarios which cover the range of these factors. This is in terms of both the time of day (shift order), and the day of the week, and are illustrated in Figure.25 which again compares the agent performance to the static behaviour case.

The agent's performance is also evaluated, in terms of its ability to generalise its action policy based on the balance of multiple constraints, without one becoming a priority. The initial training case already suggests that the intelligent agent can minimise disparity, whilst maintaining the demands of production. To stress-test this ability and validate the robustness of the approach to different conditions, the agent is tasked with an increased workload in the

same scenario. The production target is increased to 80 per hour or a nominal cycle time of 45s. The results of this simulation are presented in Figure.26.

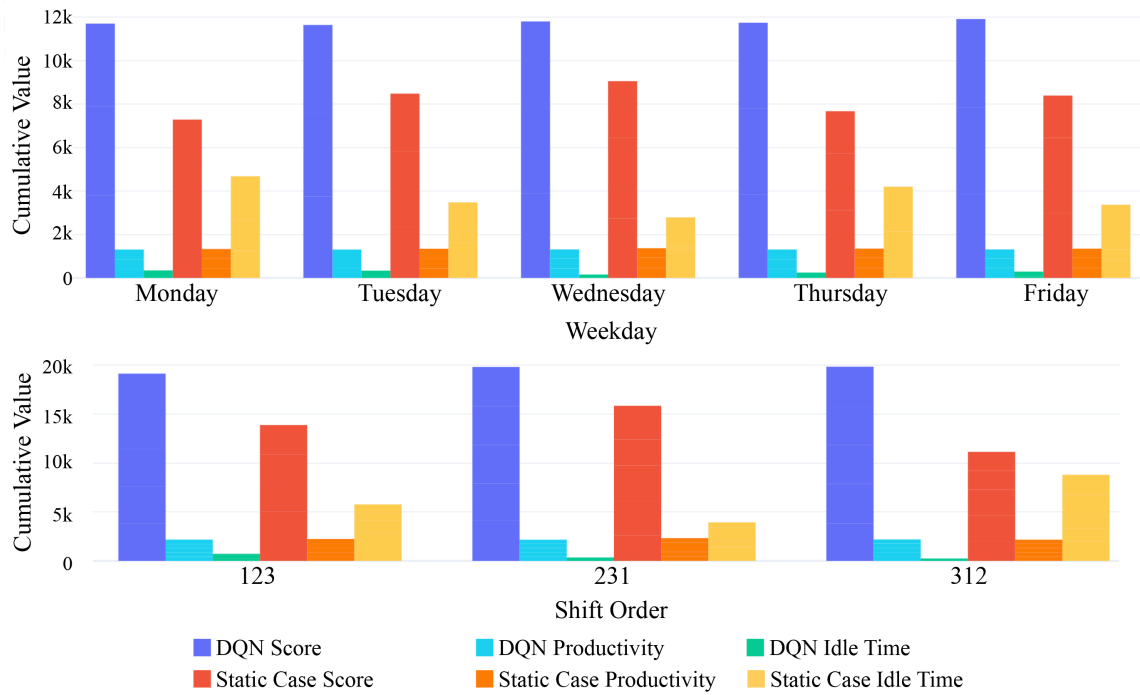


Figure 25 - Comparison of DQN and Static Case performance total over full 3-week simulation run incorporating different shift orders.

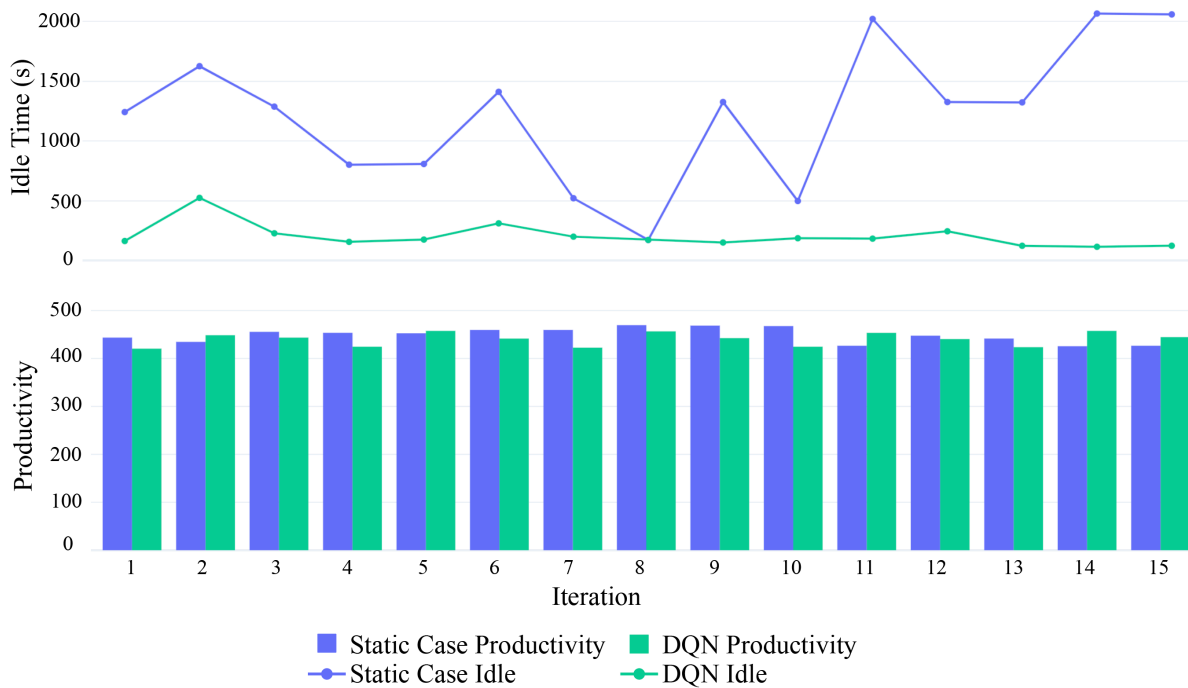


Figure 26 - Comparison of DQN to benchmark (static case) for increased production demands.

The results presented in this section detail the training and final stages of heuristic development of the intelligent agent and methodology for implementation. The results show that the reinforcement learning policy converges to an optimal solution within the simulated interaction, and has potential realisable benefits to the process in terms of cumulative small reductions to observed idle times whilst maintaining consistent productivity; supporting the hypothesis of this work. The following section expands this simulation model and the application of the agents to multiple operators and increased simulation fidelity.

8.3. Production Process Evaluation

This section covers the expansion of the case study beyond the initial interaction to encompass different interaction dynamics which arise as a result of different positions of, and combinations of demands placed on the robotic operator, and its intelligent agent. This section focuses on developing the downstream intelligent agent discussed in the previous section, further improving the fidelity and versatility in the modelling of the manufacturing interaction; and evaluating the methodology and simulation model with respect to real-world human data and leveraging prediction of human task performance from contextual factors.

8.3.1. Multiple Agents & Collaborators

As discussed in the previous section, the case-study process consists of multiple robotic operators, in both upstream and downstream positions from their human collaborator. As such, evaluation of the agent's performance in terms of its ability to enable *collaboration* at the system level. This requires the agent to learn appropriate behaviours in terms of both upstream, and downstream implications. Initially, the interaction is modelled to include the upstream human operator, and the downstream robotic collaborator. The training of this agent, using the same methodology as with the upstream case, -and the identified

hyperparameters- is illustrated in Figure.27a. Furthermore, a static case was also simulated for this interaction, to provide a comparison for the agent’s performance. This is evaluated again using the same method as in the upstream case. The results of the downstream agent’s performance compared to the static case is illustrated in Figure.27b.

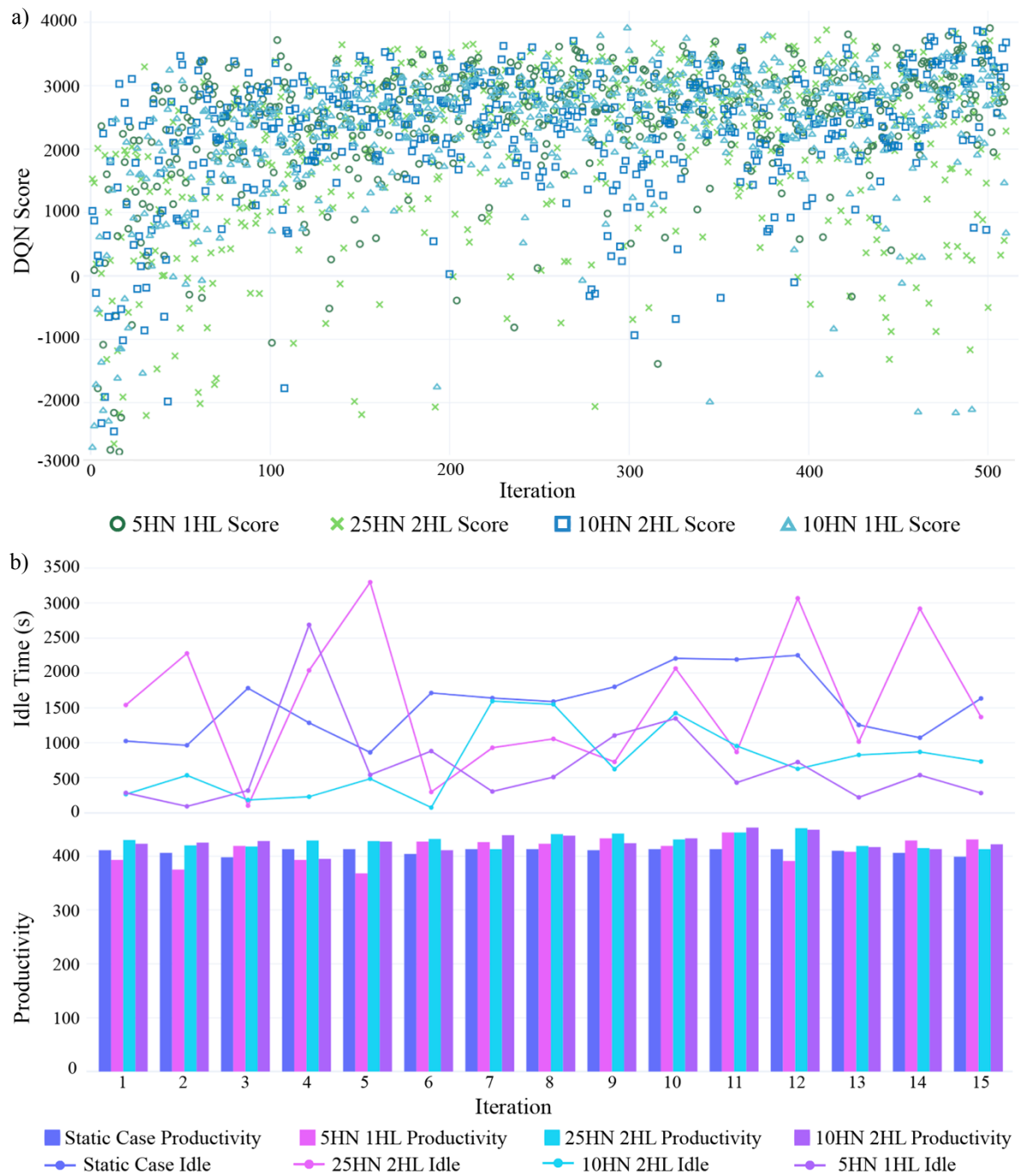


Figure 27 - a) Training over 500 iterations for the downstream operator, and b) Agent performance over the 3-week simulation run compared to static-case performance.

The presented figures illustrate that the intelligent agent is again able to determine an appropriate behavioural policy by which to minimise the performance disparity and its impact on the manufacturing process, as evidenced by the lower observed idle times and increased productivity for most of the evaluated configurations. The figures do highlight an overall lower training performance, with a looser convergence and overall score, in terms of both rewards received and the evaluation metrics.

The inclusion of this downstream operator then enables the full interaction to be modelled, and intelligent agents demonstrably capable of appropriate behaviours can be implemented in both upstream and downstream cases. This simulation configuration is run, including a static case for both operators simultaneously. The results in terms of both agents score during training, and their performance during evaluation compared to the static-case performance, is illustrated in Figure.28 a) and b) respectively.

The figure shows that in the expanded simulation model, both agents are again able to find an optimal solution for their behaviours and are able to reduce both the observed idle time and provide a modest increase to productivity. These benefits are additionally compounded by the presence of two adaptive systems. Additionally, what can be seen is that in the case of simultaneous training, the downstream operator demonstrates improved performance, in terms of both absolute score and convergence. This is likely the result of stability from iteration to iteration, provided by the upstream operator. In any sequential process, the productivity is (usually) limited by the output of the initial operator involved, who is responsible for determining the arrival rate for the process. In the case of training the downstream agent alone, this was the human operator and the variance in output that arises from the performance variation introduced a training instability. In the simultaneous training case, the upstream operator is capable of learning a policy and providing a more consistent level of productivity over different scenarios which may affect human behaviour.

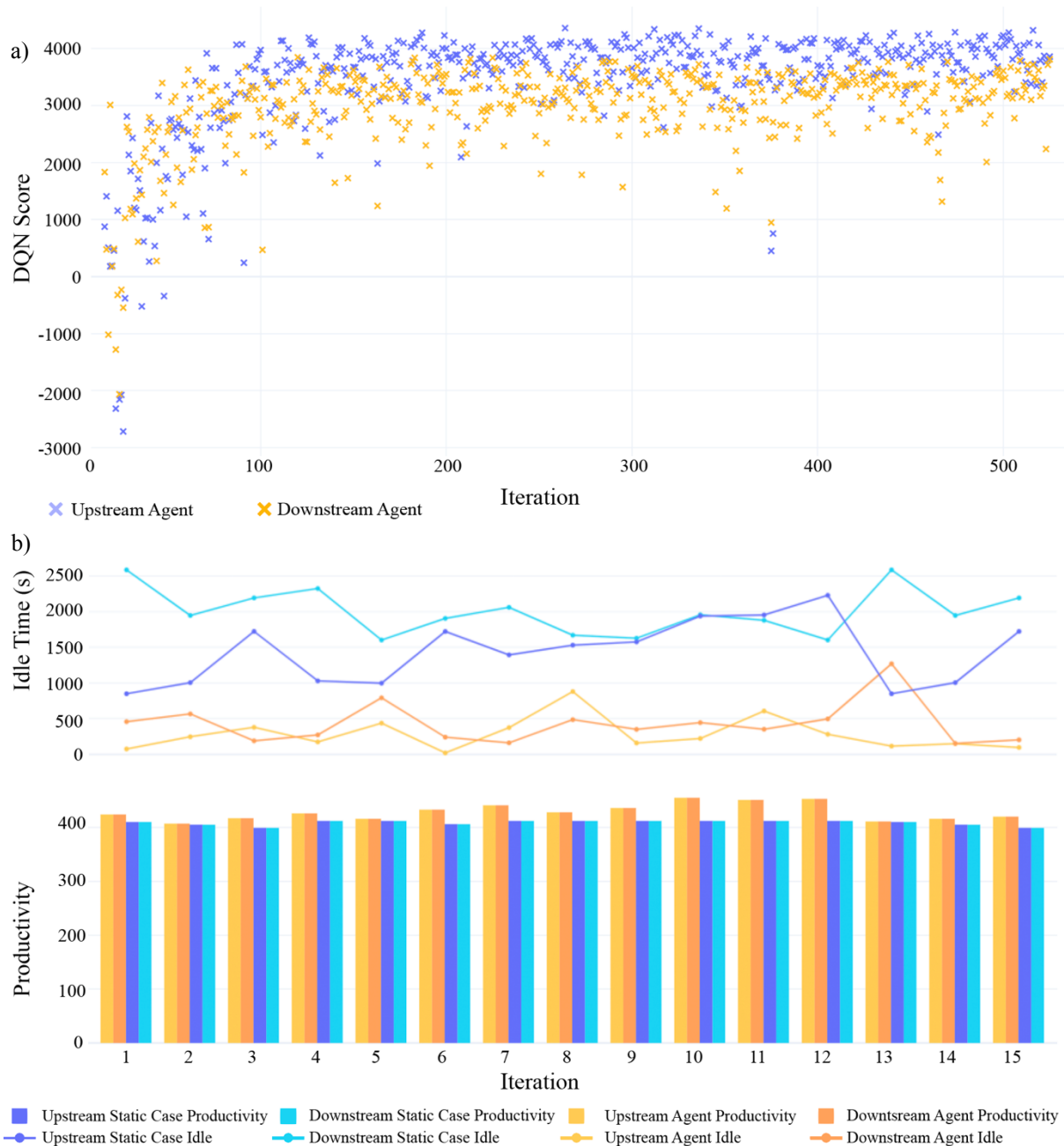


Figure 28 - a) Training over 500 iterations for both the upstream and downstream operators, and b) Agent performance over the 3-week simulation run compared to static-case performance.

Furthermore, to further evaluate the capabilities of the intelligent agent, and the application methodology, an additional case is considered whereby a robotic operator must coordinate with multiple human collaborators. This introduces additional complexity into both the dynamics of the system and consequently, the observation space and the reward

structure of the agent. The simulation model is expanded as described in the methodology, and the observation space and reward function amended to include observations of multiple collaborators. The agents were trained using the existing identified configurations. These are not necessarily optimal, but the interaction in terms of just the upstream or downstream is the same as the previous application, and likely performs well. The results of training and evaluation for the multiple human operator case are shown in Figure.29.

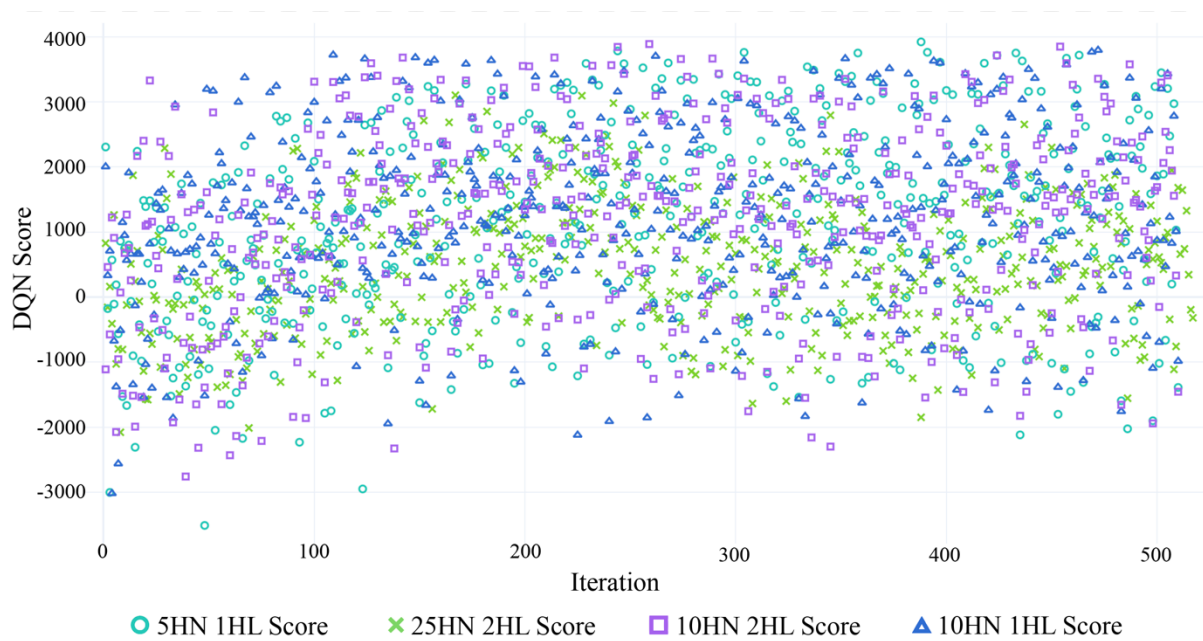


Figure 29 - Training over 500 iterations for the agent with multiple human collaborators.

The figure presenting the scores for the different agent configurations for each training iteration shows that there is limited if any, convergence to an optimal behaviour policy for the two-human case. This may be the result of increased situational complexity, and the inability of the neural network to make accurate predictions, or the variability between operators over which the robotic operator has limited influence, may prevent the existence of an optimal policy at all. This is discussed further in a later section.

8.3.2. Human Data and Performance Prediction

This section details the two final sets of evaluations. Firstly, the extension of the human elements within the simulation model, to use distributions to generate cycle times, based on collected values enabling individuals to be profiled in terms of task performance. This will further enable the robustness of the intelligent agent, by further evaluating performance in a scenario that captures the random nature of human performance. Secondly, the results of using the analytical module to predict human task performance, which has implications for both the agent's ability and the ethical constraints of data-monitoring.

In the case of capturing real-world human task performance data, 5 individuals were profiled, as discussed in the methodology and extended in the case-study section. Of these 5, Operators 1, 2, and 4 are selected to replace Operators 1-3 in the existing model respectively, due to the similarities between their nominal cycle times and variances with respect to the average time of the operation. The intelligent agent is re-evaluated using the same methodology as before, with the profiled behaviour of the human agents. The results of the DQN training, and the performance compared to the static case, are shown in Figure.30

Figure.30 follows the pattern of previous results and further confirms the agent's ability and consequently it's robustness in terms of developing an optimal solution and improving the process with respect to the key metrics. In almost all of the evaluation cases, the intelligent agents can be seen to reduce the cumulative observed idle time and leverage a modest increase in throughput. Training of the agents overall can be seen to converge, but the observed range of scores over the training iterations is much greater, possibly the result of the increased range and degree of variance in observed human task performance.

The final stage of evaluation for the methodology and agents is the combined use of both networks. Initially, the analytical network is used to provide collaborator cycle time predictions for the current time step, which are provided to the reinforcement network as part

of the state observation. This configuration has implications for applications and modelling without the reliance on continual monitoring, and to avoid direct observation and profiling of human data. Before the possible benefits of the analytics module can be explored, it is additionally important to establish equivalent performance to utilising direct observation. The results are presented in Figure.31, in terms of both the training of the intelligent agents and the predictions of the analytical network, compared to the observed human cycle times.

Figure.31a) illustrates the scores achieved by each agent during the training process. Compared to the previous simulations, both agents can be seen to converge slower, due to the secondary degree of training occurring with the analytical network. Despite this instability, both converge to a solution within 500 iterations. This solution remains unstable, however, likely due to the predictive errors of the analytical networks, which whilst small and comparable to the training case presented previously, are non-negligible, and should be acknowledged of a potential source of disturbance to the agent training and performance. This can be seen in the offset of the predictions of the analytical network in Figure.31b). This may be alleviated by further training, or different network structures, generating a higher degree of accuracy in task performance predictions, to reduce the noise which the decision-making network has to deal with.

The results presented in this chapter support the key hypothesis of this work, that a reinforcement learning based *intelligent* agent may be used to develop a behavioural action policy for robotic operators in the manufacturing setting that improves the fluency of their interactions with their human counterparts. Across almost all tested conditions, the intelligent agent was able to reduce the cumulative observed idle times whilst maintaining the desired throughput of the production process. This can be seen to provide tangible benefits at the process level over time, and the improved interaction fluency inherently makes these actions more collaborative in nature.

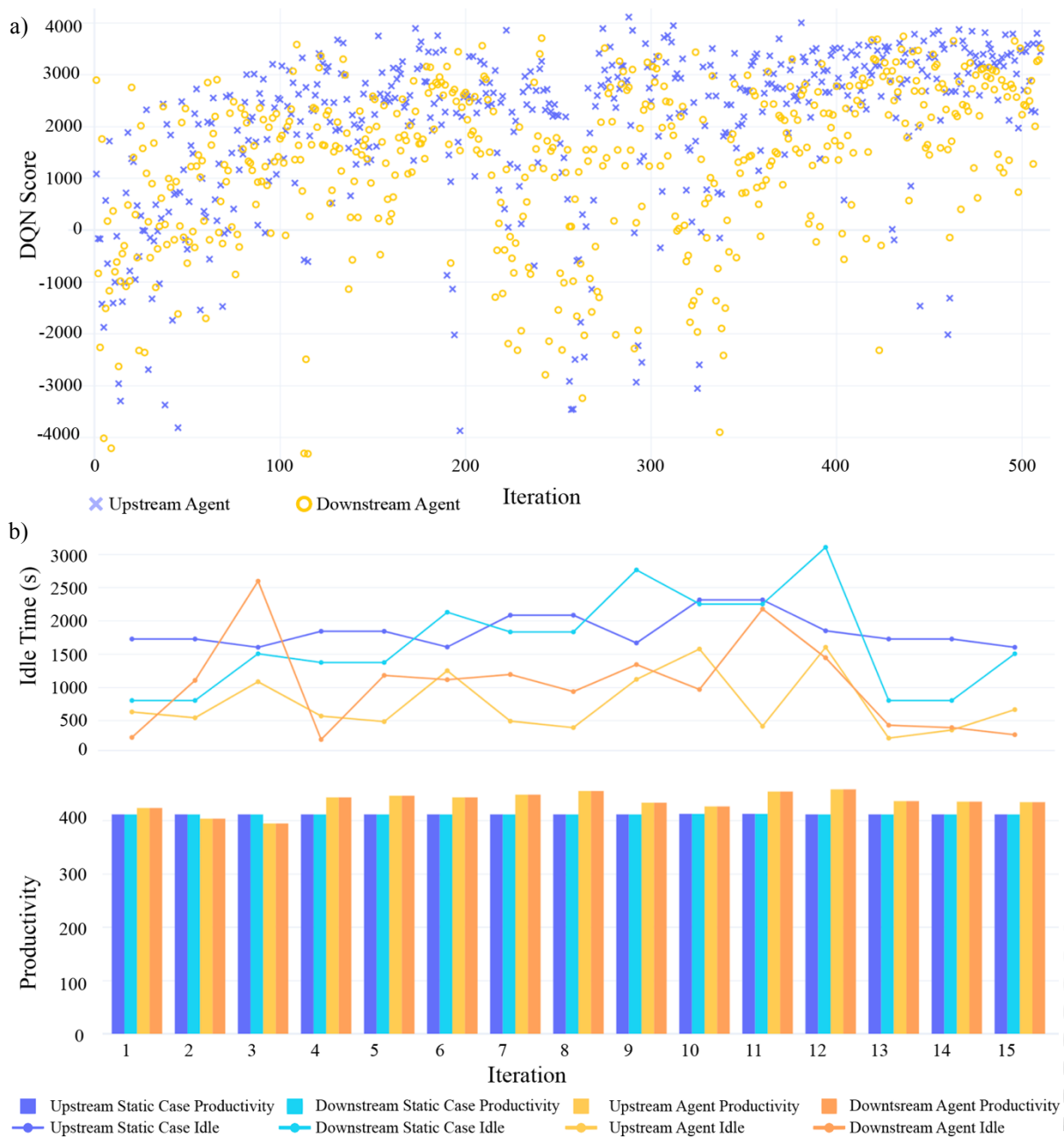


Figure 30 - a) Training over 500 iterations for the agent with profiled human task-performance, and b) evaluation compared to static-case.

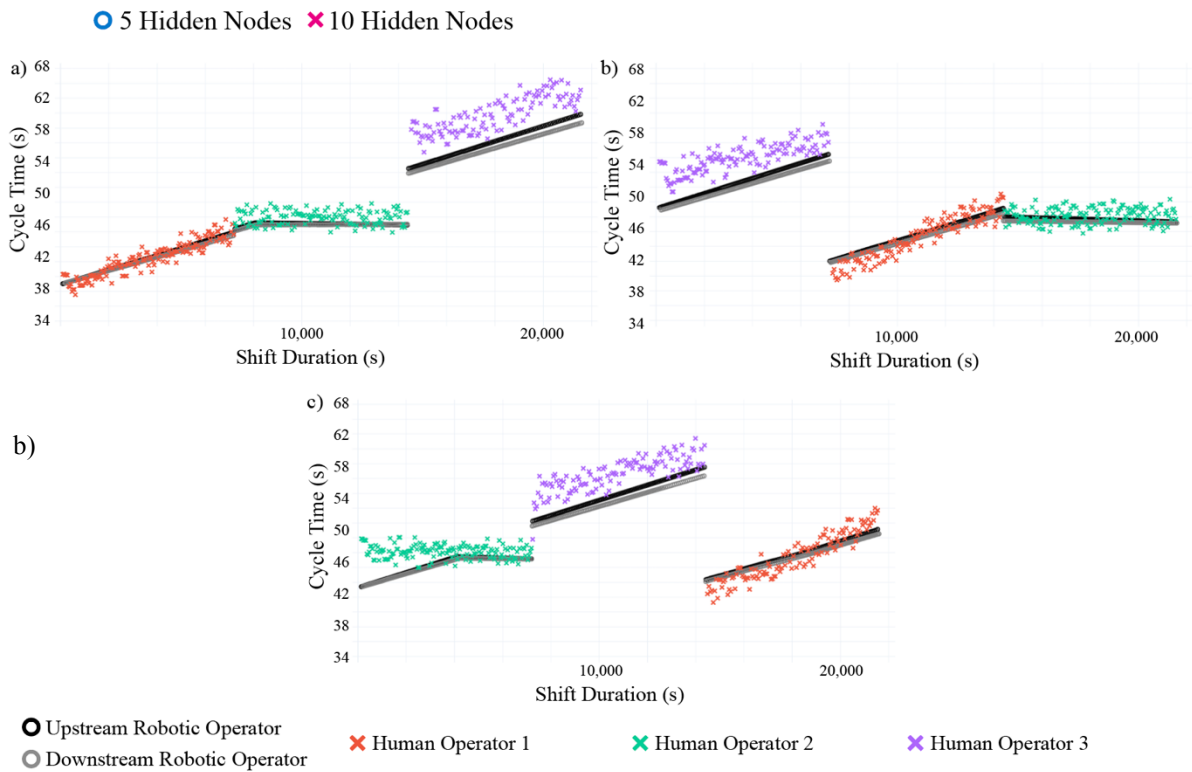
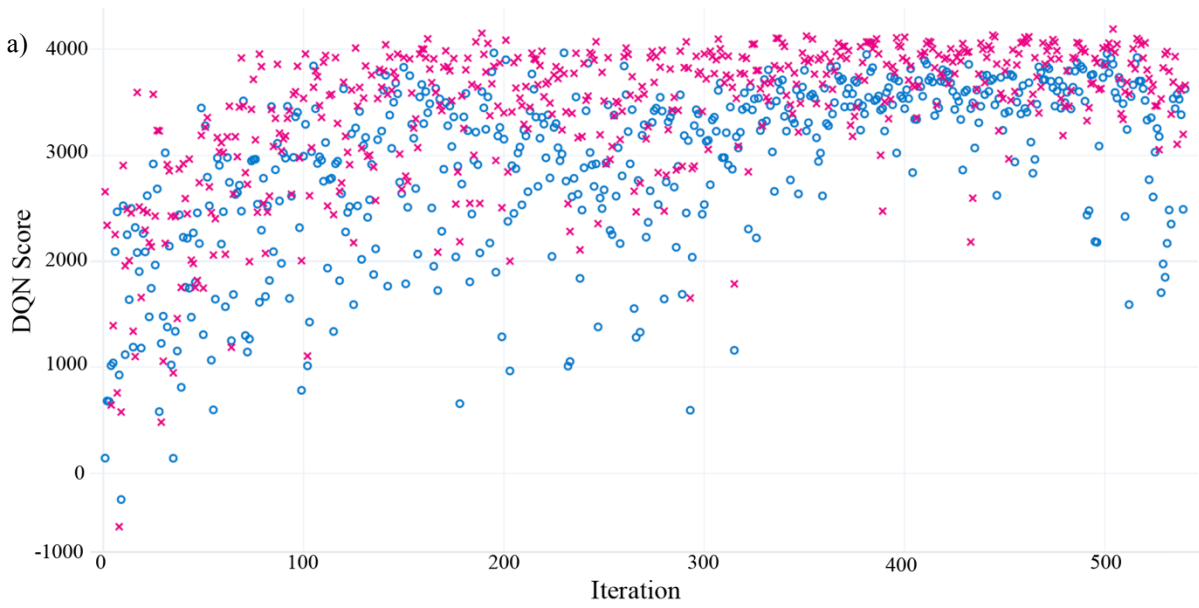


Figure 31 - a) DQN Scores over 500 iterations for the decision-making network, and b) Predictions vs observed human cycle times for analytical network for Monday of each week.

9. Discussion

To reiterate, the mass-adoption of automation technologies has, in the past several decades, led to an increasing number of interactions between human beings and robotic operators, within production processes. These production lines are designed around and increasingly aim to improve repeatability and predictability, two qualities for which human beings are not renowned. This presents several issues therefore, with achieving optimal behaviours for robotic operators within these systems. Concurrently, the increased capability of computation, in terms of both hardware, software, and design knowledge, has enabled analytical systems to be developed in related domains, capable of human-level performance over a variety of tasks, including control and prediction.

This work has sought to apply these concepts of intelligent manufacturing, to the development of a software agent which enables a robotic operator to behave in an optimal manner, in terms of improving its relationship with its human colleagues, and the production process as a whole. Three main aims were identified from the research. Firstly, to further understanding of these human-robot-interactions, in terms of how they influence system dynamics, and how these interactions may be generalised and modelled to leverage digitalisation approaches. Secondly, to explore the application of reinforcement learning to the challenge of robotic control, concerning adjusting behavioural parameters to improve social collaboration, and finally, to leverage this new understanding, into an applicable methodology and set of software, by which to realise the benefits of adaptive control, within a real-world implementation.

9.1. Intelligent Agent Development

This section discusses the finding of this work in relation to the third of the identified research questions, which sought to reconcile and apply many of the emerging technologies

in the field of intelligent manufacturing into a cohesive method and set of tools to enable adaptability of robotic operators in response to changes in the behaviour and performance of their human collaborators. This adaptability was identified in the literature as a key enabler in realising many of the benefits of implementing intelligent manufacturing techniques and is crucial to realising CPS's (Wang, Törnngren and Onori, 2015; ElMaraghy and ElMaraghy, 2016; Lee, Bagheri and Kao, 2015). Within the scope of this work, and from the perspective of process engineering, the adaptability was focused on reducing the impact of disturbance introduced into automated processes by these human operators. This disturbance introduces idle times, increased level of WIP, and a general divergence from lean operation, into otherwise well-optimised systems.

A novel solution based on the principles of distributed control to divide complex environments and problems into simpler tasks was developed, alongside several software agents; which enable the neural networks to act as an intermediary processing layer in terms of information through digitised systems and facilitate their interaction with physical systems. This layer builds on established architectures in the fields of cognitive processing and human-robot-interaction (Baker and Tenenbaum, 2014; Laird, 2012; Isla *et al.*, 2001). and provides the necessary functionality to expand these dated approaches to the capabilities of modern computational systems. Additionally, a simulation platform was identified, and a methodology defined to discretise and effectively model both the physical and human elements of these processes.

The presented findings support the efficacy of the developed method, with almost all of the trained implantations capable of learning and applying an appropriate action policy, to reduce the overall performance disparity between themselves and their human collaborators. Importantly, this is achieved whilst maintaining, and in many cases exceeding the level of production seen in the static case and resulting in an appreciable decrease in observed idle

time. The presented results demonstrate that the implementation of adaptable control with respect to changes of the cumulative value of observed idle time is decreased significantly from the static control case, providing tangible cost savings, whilst improving the fluency of the interactions of robotic elements with other agents (both human and robotic) in the system. Whilst for each individual interaction, the inefficiency eliminated from the process may be of the order of a second, when multiplied by the number of repeat interactions and scaled across a company's manufacturing operation, the cumulative effect can be considerable, and the benefits increasingly significant.

The findings demonstrate that adaptability in response to human variation has the potential to be usefully leveraged by reducing the disparity in performance. The demonstrable success of the intelligent agents at this task supports the authors' hypothesis that the integration of intelligent manufacturing concepts may be used to provide tangible benefits to manufacturing processes, both individually, and at the system level. The results further demonstrate that such an approach is well suited to aiding collaborative task performance and how these benefits can be realised using simple, proven methods and available hardware.

Furthermore, the agent can achieve an appropriate policy after relatively few training iterations and demonstrates the effectiveness of DQN's being employed for adaptable robotic control. This is additionally true for most of the presented cases and evaluation scenarios, lending further support to the robustness of the approach in terms of enabling implementation and realising the development of these systems.

The developed agents are capable of finding an optimal policy under almost all of the evaluated scenarios, and similarly in most cases, are able to affect appreciable benefits in terms of the key identified metrics. It is also worth highlighting the fact that either downstream or upstream adaptive behavioural policies can be achieved, expanding the potential applications of the work to cases with various operator configurations. Additionally,

the use of multiple agents can be seen to add stability in terms of training and performance (Figure.27 vs Figure 28). This supports the arguments from the literature review that the distribution of the control problem to multiple agents may enable optimal solutions to be achieved through division of complexity (Almada-Lobo, 2016; Leitão, 2009; Leitao *et al.*, 2016). This further suggests that implementation of a multi-agent approach into real-world processes may in some ways have advantages in terms of training and performance of intelligent systems, as opposed to digitalising and improving the intelligence of constituent processes on an individual basis.

Certain scenarios, in particular those involving multiple human collaborators, still pose issues for the developed agents in terms of finding an optimal behavioural policy. Similarly, to the static case, the intelligent agent is often bound in terms of performance by external conditions, and the variation between different combinations of these conditions may prevent the Agent from achieving a policy that is repeatable or achievable in all of the states it encounters. This can be exemplified by considering the instability in training the downstream case, whereby the agent is constrained in terms of how fast it can complete products and increase its score by how fast upstream collaborators deliver products. This is compounded when considering the need to balance multiple demands, introducing further instability. However, the generalisation approach to modelling the interactions potentially enables the same process to be divided in different ways, and these tasks re-distributed to different agents. In the case of multiple operators, this may involve tasking each agent with reducing the disparity between only itself and its downstream collaborators, which may lead to smaller individual, but still significant cumulative improvements. The ideal distribution of tasks and constraints to individual agents within simulation models is a potential area for further exploration.

Regarding the implementation into real-world processes and the realisation of the benefits proposed by this work, the generated solution provides a validated method and set of tools to do so. The presented work includes numerous enabling technologies under intelligent manufacturing, and in addition to providing a methodology to develop and implement intelligent agents, it provides insights into effective digitalisation of these processes. The platform developed to contributes to understanding and best practise of how simulation can be used to explore the efficacy and implementation of intelligent systems. This has particular relevance when dealing with learning algorithms, as a representative simulation model based on a digitalised process enables functional, application-based testing in a controlled environment, speeding up the development process, improving the performance of these algorithms, and facilitating real-world implementation.

The inclusion of intelligent software agents within simulated environments is a natural extension of existing technology and methods (see section 2.2). Discrete event simulation models are commonly used within process engineering, as they enable environments to be designed and built to replicate any number of manufacturing processes and scenarios. They are frequently employed to enable informed decision making from the perspective of management, and the expansion of these simulations to full digital representations of real-world processes is a key identified aim of intelligent manufacturing (Kritzinger *et al.*, 2018; Erol *et al.*, 2016; He and Bai, 2019; Bochmann *et al.*, 2017). This work contributes largely in this area, by formalising a methodology for the generalisation of manufacturing processes and interactions to enable effective simulation, and the development of bespoke machine learning solutions which can be evaluated in a representative, yet isolated test environment. The process of generalisation is intended to enable maximum applicability of the approach to scenarios where robotic operators are responsible for the manufacture of products with human collaborators, and provides a solid foundation for understanding these interactions, the

necessary information which must be monitored, and the development of adaptive control policies for the robotic elements involved in these interactions.

In addition, the software developed during this research (in terms of the Java code written to form the software object which performs the algorithms processes within the simulation (See Appendix.1)) can be used to implement and experiment with reinforcement agents through an MDP, in a customisable environment, that can provide a representative task for training and evaluation. Despite the strengths of the chosen simulation platform, any appropriate software package capable of parsing or with API's to convert Java can be used to experiment with these agents. This is another contribution of this work and extends the existing work in terms of usable applications by providing a useable platform, improving the scope for the application of reinforcement learning technologies into manufacturing systems, and the adoption of the methodology in practise both in industry, and for further research.

The contribution of the work in terms of supporting further research is important, as whilst the proposed approach is intended to generalise in a way that is representative, and is demonstrably applied to a case-study, fully validating such an approach will require work in considering a wider range of tasks and scenarios that may be encountered over a variety of different applications and real-world case studies. The standardisation of approaches remains a key challenge in terms of enabling the adoption of intelligent manufacturing techniques. Such standardisation is likely to remain challenging, as the process of development presented in this work has demonstrated, in almost all solutions a bespoke selection of network structures, hyperparameters and rewards are necessary to achieve optimal functionality. Additionally, and within the scope of leveraging intelligent systems to improve systems with a human component, the degree of variation between individual human beings, under different circumstances, and over a wide potential number of different manufacturing scenarios compounds the difficulty of such a validation task.

To summarise, the work highlights the value of pursuing adaptability in robotic control systems, improves knowledge on machine learning and principles of intelligent manufacturing can be leveraged within production processes. Utilising learning algorithms to provide robotic operators with agency, and tasking them with reducing the disparity in performance between themselves and their collaborators can be seen to reduce the observed idle time within these demands and levels of WIP; moving these processes closer to idealised one-piece-flow improving their efficiency and lean operation in line with the aims on many contemporary process management theories.

9.2. Intelligent Agent Performance

This section discusses the findings of the work relating to the second of the identified research questions, namely focused on the application of intelligent methodologies to the problem of interactions between robots and their human counterparts. Based on existing applications in similar domains, this was done using a multitude of neural networks, which have proved again to be versatile in terms of their application, due to their ability to abstract a range of input variables to multiple desired outputs. The presented work sought to demonstrate the application of neural networks within the context of manufacturing systems, both in terms of identification and development of the algorithms themselves, and appropriately modelling the operations, scenarios, and interactions, in such a way as to enable effective learning.

From the perspective of neural network development, there were two key aspects to the question covering two different applications. The first, focused on developing a neural network model to predict the impact on task performance of several human factors, identified from the literature (see section 2.5), and anecdotal human experience. Development of effective machine learning methods to realise CPS's behaviours is a key challenge in their

realisation (Monostori, 2014; Spezzano and Vinci, 2015), and the work contributes largely in this area. The presented methodology provides enables the identification of relevant human factors within these interactions, and how to capture and construct this information to enable a neural network to make accurate predictions of their influence on human behaviour. The results presented in Figures.20 & 21 illustrate that networks lacking in complexity are able to achieve a good level of predictive accuracy, in terms of this variation. This has important applications in terms of predictive analysis and modelling of human-involved systems, as there are limited methods to provide effective modelling of the variable and random aspects of human task performance. Furthermore, the ability of the network to effectively analyse contextual and environmental factors enables modelling and prediction in unknown scenarios and in instances where continuous monitoring is not a possibility.

The second of the key aspects concerning the application of intelligence is the application of a Q-learning approach to the control problem concerned with reducing the performance disparity and consequently improving the human-robot-interactions. Use of neural networks as value-estimators in Q-learning has been well documented in various domains, both theoretical and practical, as discussed extensively in the literature review of this work (Watkins and Dayan, 1992; Mnih *et al.*, 2013; van Otterlo and Wiering, 2012; Sutton and Barto, 1998), and the presented solution supports the hypothesis that the approach may prove useful in terms of enabling intelligent manufacturing and its constituent technologies to be realised. Achieving this required the development of a neural network, and method for implementing this algorithm within the interaction environment, in such a way as to effectively train it using an MDP. The developed methodology provides a validated process to define the appropriate observations, and appropriately discretise a robotic operators action space, to enable the network to learn an appropriate behavioural policy.

This is supported by the initial development and training of the intelligent agent, illustrated in Figure.22, which demonstrates that in terms of cumulative reward, the approach enables the agents to effectively learn a policy to maximise their scores within the simulated process. The absolute value of these scores is arbitrary, and dependant on the values chosen for the application, but the distribution of how they are earned over time is important to assess the efficacy and learning of the network. The pattern of these scores follows that expected following an approach leveraging epsilon decay during training, which the results lend further support to the efficacy of. The scores can be seen to have an initially random distribution, as the initially high epsilon value encourages exploration of the state space. These scores can be seen to converge over time to a consistent value, as the network learns optimal actions for observed states, and the epsilon value decreases, passing more control to the network. This pattern in the distribution of these scores over time is indicative of sufficient optimisation and is repeated over most of the other applications tested, lending weight to the robustness of the approach in terms of learning an effective behavioural policy. The neural network converges to a solution in the majority of applications, and generally less complex network architectures can be seen to lead to more stable learning performance.

These remain variations in the DQN scores could be further refined through additional training or may well be the result of the constraints imposed by the system dynamics; this concept is discussed further in the context of training the neural network algorithm to collaborate with two human operators.

In addition, the work establishes a method for providing rewards in such an environment, which is crucial in enabling an effective learning policy to be established. This is highlighted by the evaluation of the different reward structures (illustrated in Figure.24), which shows a significant disparity in the performance of the neural networks between the penalty and scaled approaches. The use of penalty-based rewards is optimal in terms of

developing a robust and effective solution within this application, (and indeed in a repeatable environment in general, whereby each state is similar to the previous one), and this work provides good evidence for the use of a fixed reward, penalty-based policy for generating rewards to facilitate improving these interactions. In a penalty-based policy, the rewards received on transitioning to a state will be similar in terms of value, and it is unlikely that one state will directly lead to a much more valuable future state. The opposite is true for the scaled policy, where the received reward becomes exponential as the performance disparity approaches zero. Furthermore, the almost irrelevance of the gamma parameter, and consequently the neural networks ability to predict the rewards of future states, is also likely due to the stability of the received rewards when utilising the penalty approach.

The robustness of the developed networks is evaluated by assessing the agent's performance over several tasks, with varied constraints. The results for the initial system-level evaluation presented in Figure.28 illustrates that the Agent is able to determine an optimal policy with respect to the received rewards and that this policy remains consistent policy across various scenarios. The results in Figure.25 explore this in terms of the interacting influence of human factors, and this is further supported by the following experiments, which demonstrate the convergence of multiple potential network configurations over a wide variety of constraints and interactions, further demonstrating the general resilience of the intelligent agent to variations in its goals and the constraints imposed by the specific task. This is perhaps best supported by Figure.26, which provides a deliberate stress test in the form of increased production demand and highlights the networks ability to learn a policy which represents a balance of the demands placed on it.

The results also allow additional conclusions to be drawn concerning the network structure, which demonstrably has a significant impact on the training and accuracy of both the analytical and decision-making networks. Throughout the experimental phase of the

research, an effort was made where appropriate, to compare existing optimal network structures in new scenarios, to establish the generalisability of individual network structures. This was inspired by the aforementioned work on *embodiment* (Young *et al.*, 2011; Hoffman, 2012), which suggests that unique behavioural policies may emerge from identical learning processes, given different cumulative experiences. Generally speaking, the findings presented support this theory to the extent that whilst some network structures are able to find a solution in multiple domains (namely, the 5 hidden-node, 1 hidden-layer configuration), there is almost always a bespoke combination of hidden nodes and layers which results in the best performance. It is also interesting that the best generally performing network architecture was, by contemporary standards, a simple model, containing only a single hidden layer. This may indicate that the network is overfitting to the training data, however, performance in previously unseen environments is generally good, and the resultant policies lead to an observable improvement in the key metrics, suggesting this is not the case. Regardless, this represents an opportunity moving forward, to expand the scope of the work and capabilities of the intelligent agent by incorporating more advanced network architectures.

Further work on more advanced network architectures should enable increasingly accurate and capable models to be developed, in line with more demanding learning tasks (Gu *et al.*, 2018; Kusiak, 2019; Chen, Ying and Laird, 2016). This includes investigations to explore the strengths of different network architectures designed to provide higher-level analytical processing, including convolutional networks, to enable information to be extracted from visual systems, and recurrent networks, which enable a short term memory, enabling the network to more accurately determine temporal patterns, with particular relevance to sequential processes, typical of manufacturing. Consequently, as proposed by many of the established cognitive architectures (Laird, 2012; Isla *et al.*, 2001; Modha *et al.*, 2011), and given the evidence presented in this work in Section 7.3.2, such an approach is

well suited to implementation in a modular fashion. It is further hypothesised that future approaches may build on the principles of combining multiple networks established in this work, to account for, and enable other intelligent functions, such as analytical thinking, memory, and perception, by leveraging these novel network architectures, in the decision-making process.

Generally, increasing the depth of the neural network in most applications resulted in no appreciable gain in either training performance or predictive accuracy, and in many cases resulted in a decreasing in the stability of training, and increased computational cost. These findings support the applicability of the approach, in terms of being computationally efficient, which represents a key consideration in distributed computing and control. Together, this evidence suggests that the network structure is the crucial factor in enabling intelligent data processing and that for most applications, bespoke development of learning models will be required, dependent on the process modelling, and the available dataset, its contents and structure.

As such, it can be concluded that the developed networks have an inherent degree of robustness and generalisability in terms of variation in the application, and the constraints of individual applications and scenarios. This research adds to the existing body of work in the area from the perspective of applying neural networks to real-world problems, and similarly proposes a novel solution to an unaddressed issue within intelligent manufacturing, by utilising a demonstrably capable method from other domains.

The application-specific nature of developing optimal solutions for a machine learning problem necessitates a certain level of iterative development to identify network structures capable of accurate and reliable prediction. This process has been well documented in this case, and whilst the presented neural networks can be producing accurate predictions and learn appropriate action policies, there remain sources of inaccuracy and instability

throughout the application of these methods. Such inaccuracies are likely to be exacerbated by the challenges and factors associated with the application to real-world processes, and the processing of real-world data, with all of the noise, missing values, and additional randomness associated with human beings to account for.

The parameterisation of real human performance and the use of simulation models incorporating this parameterisation method to model human performance provide an exploration of the impact that the increased variation and noise introduced by real data has on the training process and the neural network's predictive ability. The results presented for this case in Figure.30 suggest that whilst there is a moderate decrease in training stability between iterations, the agent is still able to resolve to an optimal behavioural solution, which again lends evidence to the robustness of the approach in terms of overcoming these challenges, and consequently, its validity in terms of applying intelligent processing to a real-world application.

To answer the identified questions, the findings of this work suggest that neural networks have considerable scope for application to decentralised robotic control systems, intending to provide them with adaptable behaviour. They represent a robust solution, and whilst a certain degree of bespoke development and optimisation is required, the work presents an effective and robust approach to applying these methods to existing systems and realising these benefits.

9.3. Human-Robot-Interaction

This section discusses the findings of this work with respect to the first of the identified research questions, regarding the understanding of the interactions between robots and their human counterparts from the perspective of manufacturing. The body of this work has been focused on the application of intelligent methods to facilitate these interactions and have

primarily developed additional insight that is useful and relevant to this field, but the work also has implications for the wider field on understanding these interactions as they exist within this context. Primarily, the work presents a case which contrasts the existing prevailing applications within the documented examples, where the focus typically exists in terms of direct interactions (see section 2.4), and the leveraging intelligence to provide physically collaborative actions, such a coordinating with, and aiding or learning from human motion, either in terms of supporting human labour, or capturing the fidelity and flexibility of movement. The presented approach proposes a novel application in terms of facilitating interactions which occur between humans and robots, on the passive level. This solution can leverage knowledge of its collaborators' performance to inform behaviour in robotic operators within the manufacturing context. The ability to respond to changes in the behaviour of others is a key ability in terms of enabling collaborative behaviours in interactions with others, and extension of this ability to make use of contextual information furthers the scope of work in this area. Specifically, the adaptability is used in this application, to reduce the disparity in performance between robots and their human collaborators, that arises from the variation in between both individuals and as a result of various factors which influence human performance.

This approach extends work in the field of social cognition, and through the developed cognitive architecture, provides a novel method for implementing knowledge of a collaborator in the form of the intelligent agent. In addition to providing an implementation of a collaborative system, the findings of this work provide higher level knowledge on how to facilitate social cognition within robotics, at the larger level, and across applications. The demonstrated success of the approach supports the use of cognitive architectures, and the proposed framework and methodology, as discussed, building on many of the established concepts within enabling social behaviours (Wiltshire, Barber and Fiore, 2013; Wiltshire *et*

al., 2016; Lemaignan *et al.*, 2017). The approach reinforces the use of modularisation as a general tool to discretise different areas of cognition and enable them to interact effectively. This is best demonstrated in the work, from the results presented in Figure.31, which documents the training of the intelligent agent using a separate network to make predictions about human performance. This importantly leads to equivalent performance as in the case of making direct observations and has implications from an implantation perspective, including data-monitoring, which will be covered later, but also explores the efficacy of isolating individual neural networks to perform different aspects of cognition.

Further to this, the work provides evidence that supports the links between many concepts of cognitive processing and distributed control. Notably, the similarity in modelling between human agents from a cognitive perspective, and learning agents acting within an environment. There are key parallels between the Theory of Mind (Goldman, 2012; Baker and Tenenbaum, 2014) which determines how humans perceive and act in response to an environment, and how agents perform a Markov Decision Process, and the constituent data processing steps involved. These similarities were initially hypothesised to facilitate a distributed approach in terms of enabling social cognition, due to these similarities in information processing, and the findings of this work support this hypothesis.

Relating specifically to the key findings and contributions concerning human robot-interaction as it occurs within manufacturing processes, there is clear evidence that enabling robotic operators to intelligently process and respond to observations can improve robotic adaptability. This adaptability can be extended to improve the fluency of the interaction between robots and their human counterparts in the manufacturing case, through a reduction in the disparity in task performance, resultantly minimising the impact of human performance variation at the system level. As such, the application of intelligence to human-machine interaction in this way contributes to knowledge that is vital in the current manufacturing

contexts, as robotic elements become more intelligent and capable; and the potential number of opportunities for them to interact with humans continues to grow (Hoffman, 2013; Wang, Törngren and Onori, 2015).

Methodologies such as that presented are further required to effectively model and processing information in these interactions, are essential to develop, in addition to the technological capability to do so. With respect to the human element of these process, this work extends the current field in terms of developing a novel approach to modelling the human elements of these systems, in such a way as to account for various human factors which are often overlooked from a process engineering perspective (see section 2.5). Furthermore, the approach also considers the capture of profiles of performance of real operators, in such a way as to enable more representative modelling, which has significant implications for cases where full digitalisation is not possible, alongside the larger questions of data privacy as a whole.

In terms of efficacy and the potential for implementation, the presented approach can be seen to effectively reduce this performance disparity between a robotic operator and its human colleagues; and this capability is repeatable over a varied set of scenarios and constraints. The improved fluency of these interactions provides benefits from a process engineering perspective, alongside softer benefits from a human management perspective for the human operators, including reduced levels of stress and a greater degree of trust with their robotic collaborators (Romero *et al.*, 2016; Hancock *et al.*, 2011; Hoffman, 2013; Lohani *et al.*, 2016).

The presented results (specifically in the case of using profiled operator behaviours) demonstrate general robustness of the approach in terms of handling real human variation, and the approach has good generalisability in terms of developing a solution. Further to this, the developed agents achieve consistent performance across both these data-driven cases and

the others, supporting the assumptions made in the simulation model and improving the validity of the approach. These factors suggest the approach may go some way towards overcoming the challenges associated with robotic systems interacting with humans in the real world. However, the dynamics of human behaviour can still be seen to cause issues for the developed agents, especially when the task extends to multiple collaborators. The two-human configuration shows considerably less convergence to a solution to any of the other simulated interactions. This is potentially the result of the increased complexity, in terms of both the observation space of the agent, and the increased number of potential penalties, but it may also again be the result of the interaction itself. The disparity between humans and their robotic counterparts is the focus of the intelligent agent, but the disparity between the two human operators may be of a range large enough that an optimal solution does not exist to average this disparity and reduce the observed idle times. (i.e. disparity is large enough that changes to suit one human immediately results in penalties from the other, and vice versa. In such a scenario, the convergence of the network is unlikely).

As the individual operator is unable to affect the human task performance, the disparity between the two collaborators is out of its control. However, human beings do have a natural instinct for reducing performance disparity, which is not factored into the modelling within this application. This is an ability that has not yet been formally studied but is best thought of in terms of trying to pass many items between yourself and another. It does not take long for a natural rhythm to form between two human individuals. With a greater understanding of this phenomenon, agent behaviour may be further improved in terms of minimising performance disparity, with multiple human collaborators.

The results support the author's hypothesis that such adaptability, leads to reduced performance disparity between operators within manufacturing scenarios, can be said to improve the fluency of these interactions. The findings provide significant contributions in

terms of the design of intelligent systems capable of social intelligence, demonstrating a level of adaptability to variation in the behaviour of others. Although the results are promising, the reliance of many of these techniques on continuous data collection poses unique issues where these systems extend to human-machine interaction

9.3.1. Ethical Use of Human Data

This section deals with perhaps, what is in my opinion, perhaps the most insightful contribution of this work. The work presented so far presents a novel solution to a problem, in such a way that may enable the realisation of tangible benefits, however, in its completion, it has posed a question that extends far beyond the scope and ambition of this work. This question is asked in light of questions and events in the economic and geopolitical climate of the last few years, surrounding the privacy and ethical use of collected data, and how it relates to human beings. These questions highlight larger issues, and their consequent application within the scope of this work, in terms of the responsibility and rights that lie with companies, and which with the individuals, when dealing with human data should be addressed. This section is not intended to be overlong, but it is important to establish the appropriate considerations so far external to the scope of this work and appropriately contextualise the issue, as to how it relates to the findings presented thus far.

It is perhaps most appropriate, to frame this in the context of a pop culture reference: *“Your scientists were too busy wondering whether they could to stop and think about whether they should”* (Spielberg, 1993), succinctly, whilst this work has presented a method by which to leverage benefit from the monitoring of human performance, it has simultaneously raised questions regarding the nature of these technologies in general, in terms of their relationships with humans, and the consequences of this fact. The discussion that follows aims to highlight the potential issues that the collection and analysis of data on such a scale and at this level of

fidelity creates, and ultimately, to encourage discussion of these points before these systems become commonplace.

The information age has led to collection and analysis of data becoming essential to the function of modern society, and yet the social, political, and economic effects and opportunities for exploitation of national and international systems of mass surveillance remain poorly understood. These opportunities for wealth creation stem from the realisation that personal data exists as a product, which can be bought, sold, and manipulated to create revenue in such a way that draw enables parallels to be drawn with traditional capitalist models. This has resulted in a new economic logic based on behavioural prediction and modification, termed *Surveillance Capitalism*, and described in the words of its architect, as: "*A new economic order that claims human experience as free-market raw materials for hidden commercial practice of extradition, prediction, and sales.*" (Zuboff, 2015; Zuboff, 2019). The work ultimately suggests that these practices present as great a threat to human nature itself, as industrial capitalism does to the environment.

The use of human-data as a resource has been leveraged to a huge extent not only within the context of manufacturing (where it proves useful in the refinement of existing processes, and the creation of entirely new, innovative revenue streams (Srnicsek, 2017)) but across a wide variety of different domains. Until recently, it was easy to dismiss the more existential threats of this paradigm, as the development of these technologies and the emergence of systems built on data have improved individual lives in many ways, including real-time traffic data and logistics information extracted from mobile phone locations (Shaw, Tsou and Ye, 2016; Guido *et al.*, 2017), and personalised shopping and media recommendations based on an individual's previous purchase, behaviours, and more recently, modelled preferences (Hallinan and Striphas, 2016).

The issue is that industry and consumer behaviour is now governed by these paradigms, with impacts that extend to the societal and cultural levels, with firms able to adopt these new practises becoming household names (Netflix, Amazon, etc.) at the expense of their competitors, further propagating these practices. The pervasive nature of these service has, however, reached a tipping point where it feels noticeably uncomfortable to many. There is much discussion, both published (Nichols, 2018; Kleinman, 2016), and anecdotally, concerning the practice of analysing audio data collected by the microphone inside mobile phones. This data enables key words and information to be extracted, through the use of Natural-Language-Processing (NLP), which is either used directly or sold to third parties for the purposes of advertising.

These concerns are further exacerbated, by the political and societal debates occurring on a global scale surrounding the fair and ethical use of data, and the rights of individuals to privacy. It is a controversial topic at the heart of several recent news scandals, involving significant geopolitical events. One of the more notable examples is the Cambridge Analytica scandal (Cadwalladr and Graham-Harrison, 2018; Isaak and Hanna, 2018), which brought into the public consciousness, and awareness of how pervasive the monitoring and profiling of individuals for commercial use has become, and importantly how targeting individuals based on the knowledge extracted from their personal data enables human behaviours to be manipulated at scale. The activities of Cambridge Analytica were a stark demonstration of how personal data can be used to manipulate behaviour at scale comes with potentially long-lasting very serious implications, including high profile political events, including the 2016 Presidential elections, and the highly controversial Brexit Referendum the same year, the effects of which are to be felt on a global scale at all levels for decades to come.

This example is more of an exception than a rule, but it sets a precedent for the appropriate use of human data in other areas (especially within a few certain domains), where

issues can already be seen to be emerging concerning how employers monitor and make decisions based on the behaviour of their employees.

A notable example of this can be seen within the US trucking industry, where the move to the use of Electronic Logging Devices (ELD's) in place of conventional analogue methods to monitor the work schedule and task completion. The case raises some interesting points regarding the efficacy of monitoring and data-based enforcement, specifically where it restricts their autonomy. The use of ELD's has the potential to cause issues as tight working regulations exist regarding the time spend driving versus sleeping. Such regulations exist within logistics industries on an almost global scale, but such issues are particularly acute in the US, where distances are vast, and shifts often last several days. American truckers often live within the confines of their vehicles, travelling with family and pets, and as a consequence, are affected to a far greater degree by workplace surveillance than most employees.

Drivers argue that the use of external devices to monitor and determine the behaviours, overriding and restrict their autonomy, and often introduces inefficiencies into their working patterns, which leave all parties involved are worse off. Drivers may find themselves a mile from their destination and have their vehicle remotely shut down and a rest period enforced even when their assessment of the safety of the situation suggests they are fit to drive the remaining mile. The consequences of this scenario being a decrease in the overall productivity of logistics systems, as a result of delivery delays, liability for overtime payments, and decreased quality of life for the drivers. This case is not alone, but perfectly illustrates one of the key issues in terms of misapplication of surveillance in terms of removing autonomy from individuals, and making decisions based on poor, or incomplete models, which fail to account for the situational context (Vox, 2017; Levy, 2015)

This discussion covers a small number of examples but stands to make the point that the collision between the competing ideologies of the human for advancement through technology, and the human desire for privacy which exists in the information age is becoming increasingly relevant. This is an eminently human dilemma, that arises from a societal shift away from humanist ideals, whereby human experience ascribes meaning and value to objects and events, to ideals based on the concept of *dataism*, which reduces the development of human consciousness to algorithmic processes, and indeed the nature of the universe to flows of information. These ideals propose, that algorithms are inherently more capable than human beings, and consequently, the valuable actions are those which contribute in the greatest terms, to these dataflows, and the knowledge and wisdom that can be drawn from them. How we reconcile these new ideals This can be argued either as a threat to humanity or simply the natural next stage of our evolution as a species. For an excellent discussion on the emergence of see (Harari, 2014).

Strong arguments exist on both sides of this division, the degree to which employers have a right to monitor their employee's vs the right employees have to privacy is an ongoing argument, as evidenced by an extensive body of literature on the topic (Martin and Freeman, 2003; Ciocchetti, 2011). These issues are made more urgent, by the ongoing societal discussion of personal privacy, which is bringing attention to the ethical and legal responsibilities of corporations and governments with respect to the responsible handling of this data. From a legal perspective, in most major jurisdictions, the employer is entitled to collect and process whatever data they deem relevant to their business operations, and such protections are necessary, with benefits to both employers and employees, in terms of work being successfully completed. Indeed, the presented works aimed to leverage machine learning in the monitoring and analysis of variations in human task performance, with the intentions of using this information to provide tangible benefits to both the process and

human beings involved. The benefits, as discussed ad nauseam up to this point, that the capture and processing of this data enables are extensive, and it hard to deny from the perspective of the greater good, that restrictions should limit the application of these technologies within a business setting.

These issues were particularly acute in terms of the application considered in this work. A significant part of enabling the use of intelligent agents in such a way involved capturing a dataset upon which a variety of learning algorithms could be trained, to identify changes in performance and appropriate actions to interact in a socially cognitive manner. The collection and compilation of the human task performance data, necessitated in this case, highlights the other side of this argument, and these issues are further amplified in cases such as these, where the data collection may enable identification of individuals, as opposed to monitoring average process outputs and statistics.

Data that identifies individuals introduces significant culpability for sub-optimal performance, as individuals are now minutely for their actions where the actual responsibility may lay elsewhere, and potentially retroactively in light of changing rules or context. Furthermore, it introduces the potential for biases and incomplete or incorrect knowledge to be incorporated autonomously into decision-making processes which may unfairly (and unintentionally) penalise specific individuals. This may be the case when minimal efforts are made understand the dynamics and variations of these processes, whether this is between individuals or conditions, and how this is reflected in the collected data, to avoid this data being used partially, or out of context.

This has considerable implications in terms of the implementation of intelligent manufacturing, in particular Cyber-Physical-Systems based approaches, as many are reliant on collecting, storing, and processing data at high and frequency and fidelity, to achieve the necessary accuracy of prediction and decision-making. These processes, in turn, generate

profiles of employee performance which may be continuous profiles of uncontextualised data for each individual describing their performance, where these systems contain human elements. Whilst there is a valid argument to be made that this is necessary to realise the benefits of learning-based and digital modelling-based approaches, this is an issue for the aforementioned reasons, as this data may be misinterpreted either maliciously or unintentionally, without the relevant context, making it difficult to determine the causality of any effects which may be observed. The presented work makes steps towards addressing some of the issues highlighted in this section, in terms of developing a method capture variation in human performance at the individual level and enable effective simulation models and intelligent agents which are non-reliant on constant monitoring and resulting in no lasting profile.

The argument, in essence, is one of Utilitarianism vs Kantianism. In terms of how one should behave, if the issue is considered from the perspective of the former, then the employer should maximise their utility by using the data that they have available if it will lead to improved productivity, the counterargument holds that intervention is not justified by the means if it comes at the expense of the rights of another individual. This discussion is not intended to reduce this argument to a debate on the level of Philosophy (Although, for an excellent example of the comparison of these two philosophies, in terms of how they influence decision-making in terms of morality, see (Sachdev, 2017), or Ethics (Likewise, for an interesting study concerning the nature of accountability in terms of the behaviour of autonomous cars, where the consequences can quite literally be life or death, see moralmachine.mit.edu, (MIT Media Lab, 2020) for some interesting social and technical research in the area).

To surmise, the modern data-driven economy promises many technologies once relegated to the realms of science fiction, alongside pragmatic improvements to everyday life,

including more efficient and sustainable processes, seamless interactions with machines and intelligent systems, improved infrastructure, transport, and medical care, greater choice and customisation, and entirely new products and experiences. The cost of many of these technologies, however, is the mass collection analysis of personal data, and the demand for and development of datasets of an increasingly high fidelity invites numerous questions about the fair and ethical use of this data; and the extent to which it will be tolerated before it becomes an invasion of privacy at the societal scale.

In more recent days, this question has been brought even further to attention, with the rise of the threat of *Deepfakes*, made using technologies very similar to those leveraged in this work, to reconstruct the frames of video footage based on existing personal image and audio data, to replace elements, such as an individual's face and voice. Such technology has the potential to disrupt the notion of identity itself, whereby convincing footage can be produced by remote and unknown individuals based on data representations of their voice and likeness. This footage can present any individual doing and saying anything, anywhere, which is a scenario with potentially devastating consequences, if leveraged by the wrong individual for the wrong reasons (Güera and Delp, 2018; Meskimen, 2019). This raises significant issues from the perspective of personal data privacy, as it additionally poses questions as to whether individuals own the rights to their personal features. This issue is perhaps most acute as in the case of the example provided, whereby some of the individuals represented are unfortunately deceased. In these cases, individuals have no way to give permissions for their likeness to be used, and no recourse if it is, for potentially disagreeable purposes.

These questions are beginning to emerge into the collective awareness, although the issue remains one of a conflict of two ideals, which is likely to affect and influence people in unpredictable and yet quintessentially human ways. This remains uncharted territory, and

there is no procedure or process to design or validate such systems or to be sure that they are making evaluations based on models based on appropriate information, assumptions, and goals. At a personal level, the significant discovery on completion of this work, is the realisation that significant and thorough reflections are necessary as a culture, to establish the limits and responsibilities of these technologies. It is hoped that this work will encourage this reflection and further insight in the near future.

10. Contributions, Conclusions & Reflections

The final chapter of this work outlines the key findings and contributions of the research, along with the main conclusions drawn and some personal reflections on the project as a whole. Each of the sections aims to reconcile the multiple points presented and provide context of the research in order to appropriately evaluate the contributions of the work, and to draw conclusions from the results and the process of obtaining them. This research has been motivated at large, by a scenario which whilst still in its infancy, and becoming increasingly prevalent, whereby robots and humans coexist and interact with one another at the individual level. This is especially relevant in the case of manufacturing, which has long been an early adopter of automation, as the industry is frequently exposed to issues experienced by early adopters of new technologies.

10.1. Contributions

This section outlines the key contributions of this work with respect to the research questions and their motivations at the start of this project. Primarily, the fundamental contribution of this work is formed by the development, testing, and validation of the methodology to discretise, and representatively simulate a human-robot-interaction, both in terms of human task performance modelling and the design of the systems dynamics, to enable the application of a reinforcement learning approach to implement behavioural adaptability. The work presents one of the first comprehensive methodologies to realise intelligent agents capable of enabling robotic systems to behave in a way that is autonomously adaptable, and to leverage this adaptability under the principles outlined by concepts such as CPS's and Industry 4.0 to improve manufacturing processes. The method developed to do so, is supported by a codebase, which extend the capabilities of a deep learning library, to provide the relevant software agents and elements necessary for them to be implemented, both in a

simulated environment for experimentation and analysis, and with the development of suitable data input/output API's, in practice. The work does not provide a one-size-fits-all solution, as the potential scope for application presents almost limitless variation. However, the presented case-study application of the methodology to a representative real-world task demonstrates the potential for application and how it may be used to realise tangible benefits within these processes.

The central findings of this work firstly provide insight into how to effectively capture and model the performance variation of human beings, and how to suitably generalise manufacturing processes in such a way as to enable effective simulation of these interactions; concurrent with a robust method for doing so in such a way as to facilitate the use of *intelligent* software agents. This is crucial in enabling the use of these technologies, by providing an appropriate platform for the work to explore the application and development of the relevant neural networks, which the findings demonstrate to be capable of achieving accurate predictions in terms of variable performance and state values. The extension to the consideration of human factors which influence performance within modelled processes also represents an alternative and more detailed approach than contemporary work. As such, a reinforcement learning approach has been demonstrated to enable intelligent software agents, to evaluate their observed environments and develop action policies to achieve their aims.

Furthermore, work provides a number of theoretical insights into other areas, with the approach validating several assumptions from existing work, including concepts from cognitive processing, such as the benefits of modularisation in terms of isolating elements of processing, and the efficacy of embodiment in terms of improving the versatility of intelligent agents. The approach also provides an important application in terms of enabling social intelligence, by demonstrating that intelligent agents can facilitate the coordination of actions with others at a behavioural level. Whilst not necessarily conclusive, the results

further support the idea that leveraging the similarity between neural networks and the way they process and learn from representations and human thought processes is a factor in their success in this application.

10.2. Conclusions

This section outlines the key findings of the work with respect to the identified research questions. Despite the intense focus on leveraging intelligent technologies within the manufacturing field, this work covers a niche problem, that has been somewhat overlooked by the existing fields of process engineering and the novel field emerging under the concept of intelligent manufacturing. The work aimed to reconcile many different approaches to develop understanding in terms of how interactions occur between humans and the robotic colleagues in these scenarios and to develop an application of these technologies. This was to facilitate collaborative and intelligent behaviours in these robots, through adaptations to their existing routines. This posed questions in terms of understanding how to effectively capture and model human performance variation, and contextual information to enable these behaviours, along with the how improving these interactions may improve manufacturing processes; the applicability and development of a neural network-based approach to analysing this data, with respect to how to structure these networks to generate the required knowledge and perform to an appropriate level of accuracy; and the effective implementation of these learning techniques, making use of decentralised control, to develop a policy for determining appropriate adaptable behaviours through reinforcement learning to achieve a goal. Together, these questions extend the domain of human-robot-interaction and collaborative robotics to encompass a new range of interactions.

Whilst, in this case, these goals were focused on reducing performance disparity, the design of the approach was intended to enable generalisation to networks structured to

analyse whichever data points are relevant to the application at hand. The general versatility of the approach illustrated in the results of this work support this approach with respect to this generalisability and robustness.

The consequences of this are considerable, as the presented solution suggests that even simplistic implementations of intelligence following the approach of decentralisation can make appropriate predictions and decisions in real-time, to provide tangible benefits to manufacturing systems. This has importance in terms of realising benefits in terms of efficiency and productivity in manufacturing but also suggests that the approach has the potential to be greatly expanded into numerous applications, and further enhanced in terms of accuracy and capability.

In addition to providing insight into the application of reinforcement learning and the development process for these algorithms concerning discretisable problems such as the one presented, the findings have a significant application in a more tangible and immediate sense, in terms of realisable benefits to the manufacturing industry. The use of intelligent agents in the proposed way can be seen across many of the evaluations, to lead to behavioural policies that reduced the observed idle time of both operators. The benefits of this are obvious in terms of cost and time savings to the company and provide a method for enabling more lean process operation. Additionally, however, the use of socially intelligent robots has significant benefits for their human counterparts, both pragmatic, by improving working conditions in terms of through stability and personalisation; but also, soft benefits, as receptive behaviours are conducive to building trust between individuals. These benefits contribute to reducing stress and building acceptance of intelligent machines.

The main limitation of the approach emerges from these evaluations, however, and it is the assumption that whilst appropriate care has been taken to build models that are representative, and inform their design and function with representative data, there is no

validation process for the approach, and the findings rely on many of the assumptions being correct. The demonstrable robustness of the approach offsets these concerns somewhat in terms of the approaches general validity and are somewhat further alleviated by considering another key insight of the work; whereby, optimal solutions for the structure and operations of these agents is likely to require application-specific development.

As discussed previously, the field of computational intelligence has seen an astonishing rate of improvement, even throughout the duration of this research. In part, this is due to increased accessibility to the necessary hardware and software, required to develop these systems. This is likely to further facilitate the development and implementation of intelligent agents, and the work contributes further in this area, through the development of a methodology and software tools, to enable experimentation with intelligent agents, in a customisable simulation environment. The work addresses several of the numerous challenges that the implementation of such systems in real-world applications presents. These include the aforementioned issues with validating these systems, but also the challenges of dealing with human data, including the noise and true randomness that only events subject to the chaos of the real world can provide. The work takes steps towards this but falls short of utilising a fully data-driven approach, both in terms of the technical challenge, but the ethical one also.

10.3. Reflections

This final section addresses some of the more personal reflections on this work as a whole, and the results and their implication. Foremost, the work has been completed during a period of time characterised by an unmatched rate of technological progress. At the outset, the technologies utilised within this work were still relatively primitive, and yet by the end, their capabilities, and importantly, the availability of access and understanding of these tools and

techniques, already far exceed the expectations and anticipated capabilities of this approach. As such, the current state of the art of these techniques at the point of finishing this work far exceed those exhibited within it, and the potential for intelligent agents to be leveraged in the context explored currently far exceeds what is covered. Such is the nature of applying contemporary computer science. That said, and in my opinion, the presented method is quite capable of enabling a level of collaboration between robotic operators, and their human counterparts, which has realisable benefits over existing monitoring and control systems, at the process and human level. One aspect worth discussing in terms of the experience gained during the completion of this work, is the skill of knowing where the relevant boundaries of the project lie. This was especially relevant in this case, with almost unlimited scope in terms of choice of learning methods, it was necessary to concentrate on developing a robust modelling methodology and software implementation, capable of providing foundations for expansion in terms of intelligence, as new methods and techniques continue to be realised.

As mentioned previously, the presented methodology stops short of implementing a fully data-driven approach to modelling. In part, this is the result of one of the larger insights, in terms of how human data should be used, not just in the context of the manufacturing industry, but at all levels of society. How is personal data defined? What rights do individuals have to their personal data? How do these rights coexist, with the moral and economic counterview of industry? This work represents a small portion of a much larger and multi-disciplinary field, focused on leveraging the benefits that data-driven systems can provide. The presented solution is one to an even smaller problem still, and yet it is one that I believe has significant implications.

References

- [1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J. and Devin, M. (2016) 'Tensorflow: Large-scale machine learning on heterogeneous distributed systems', *arXiv preprint arXiv:1603.04467*.
- [2] ABB YuMi IRB 14000 Collaborative Robot: ABB. Available at: <https://new.abb.com/products/robotics/industrial-robots/irb-14000-yumi> (Accessed: 23/03/2020).
- [3] Adams, M., Schroer, B. J., Gunter, L., Schroer, B. J. and AL Arsenal, R. (2001) 'Impact of process variability on lean manufacturing systems'. *Proceedings of the Huntsville Simulation Conference*.
- [4] AG, K. (2020) *human-robot-collaboration (HRC)*: KUKA AG. Available at: <https://www.kuka.com/en-gb/future-production/human-robot-collaboration> (Accessed: 23/03/2020).
- [5] Agravante, D. J., Cherubini, A., Bussy, A., Gergondet, P. and Kheddar, A. 'Collaborative human-humanoid carrying using vision and haptic sensing'. *Robotics and Automation (ICRA), 2014 IEEE International Conference on*: IEEE, 607-612.
- [6] Ali, S., Al Balushi, T., Nadir, Z. and Hussain, O. K. (2018) 'Distributed control systems security for CPS', *Cyber Security for Cyber Physical Systems*: Springer, pp. 141-160.
- [7] Almada-Lobo, F. (2016) 'The Industry 4.0 revolution and the future of manufacturing execution systems (MES)', *Journal of Innovation Management*, 3(4), pp. 16-21.
- [8] Anderson, J. R. (1996) 'ACT: A simple theory of complex cognition', *American Psychologist*, 51(4), pp. 355.
- [9] Atzori, L., Iera, A. and Morabito, G. (2010) 'The internet of things: A survey', *Computer networks*, 54(15), pp. 2787-2805.
- [10] Auschitzky, E., Hammer, M. and Rajagopaul, A. (2014) 'How big data can improve manufacturing', *McKinsey & Company*.
- [11] Babcock, B., Babu, S., Datar, M., Motwani, R. and Widom, J. 'Models and issues in data stream systems'. *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*: ACM, 1-16.
- [12] Bagheri, B., Yang, S., Kao, H.-A. and Lee, J. (2015) 'Cyber-physical Systems Architecture for Self-Aware Machines in Industry 4.0 Environment', *IFAC-PapersOnLine*, 48, pp. 1622-1627.
- [13] Baker, C. L. and Tenenbaum, J. B. (2014) 'Modeling human plan recognition using bayesian theory of mind', *Plan, activity, and intent recognition: Theory and practice*, pp. 177-204.
- [14] Baldi, P. and Hornik, K. (1989) 'Neural networks and principal component analysis: Learning from examples without local minima', *Neural networks*, 2(1), pp. 53-58.
- [15] Bannour, F., Souihi, S. and Mellouk, A. (2018) 'Distributed SDN control: Survey, taxonomy, and challenges', *IEEE Communications Surveys & Tutorials*, 20(1), pp. 333-354.
- [16] Bellman, R. (1957) 'A Markovian decision process', *Journal of mathematics and mechanics*, pp. 679-684.
- [17] Benedetti, A. A., Diefendorff, J. M., Gabriel, A. S. and Chandler, M. M. (2015) 'The effects of intrinsic and extrinsic sources of motivation on well-being depend on time of day: The moderating effects of workday accumulation', *Journal of Vocational Behavior*, 88, pp. 38-46.
- [18] Bernstein, D. S. and Zilberstein, S. 'Reinforcement learning for weakly-coupled MDPs and an application to planetary rover control'. *Sixth European Conference on Planning*.

-
- [19] Berthoud, H.-R. (2007) 'Interactions between the “cognitive” and “metabolic” brain in the control of food intake', *Physiology & behavior*, 91(5), pp. 486-498.
- [20] Blake, M. (1967) 'Time of day effects on performance in a range of tasks', *Psychonomic science*, 9(6), pp. 349-350.
- [21] Bochmann, L., Bänziger, T., Kunz, A. and Wegener, K. (2017) 'Human-robot collaboration in decentralized manufacturing systems: An approach for simulation-based evaluation of future intelligent production', *Procedia CIRP*, 62, pp. 624-629.
- [22] Brettel, M., Fischer, F. G., Bendig, D., Weber, A. R. and Wolff, B. (2016) 'Enablers for Self-optimizing Production Systems in the Context of Industrie 4.0', *Procedia CIRP*, 41, pp. 93-98.
- [23] Brettel, M., Friederichsen, N., Keller, M. and Rosenberg, M. (2014) 'How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 Perspective', *International Journal of Mechanical, Industrial Science and Engineering*, 8(1), pp. 37-44.
- [24] Brown, S. and Woods, A. 'An Operations Management Perspective on Collaborative Robotics'. *Proceedings of the International Annual Conference of the American Society for Engineering Management.*: American Society for Engineering Management (ASEM), 1-8.
- [25] Cadwalladr, C. and Graham-Harrison, E. (2018) 'The Cambridge analytica files', *The Guardian*, 21, pp. 6-7.
- [26] Campbell, S. S. and Dawson, D. (1990) 'Enhancement of nighttime alertness and performance with bright ambient light', *Physiology & Behavior*, 48(2), pp. 317-320.
- [27] Cao, Y., Yu, W., Ren, W. and Chen, G. (2013) 'An overview of recent progress in the study of distributed multi-agent coordination', *IEEE Transactions on Industrial informatics*, 9(1), pp. 427-438.
- [28] Chen, C., Ying, V. and Laird, D. (2016) 'Deep q-learning with recurrent neural networks', *stanford cs229 course report*, 4, pp. 3.
- [29] (2015) *KERAS*. Available at: <https://keras.io/> (Accessed: 04/03/2020).
- [30] Chukwuekwe, D. O., Schjølberg, P., Rødseth, H. and Stuber, A. 'Reliable, Robust and Resilient Systems: Towards Development of a Predictive Maintenance Concept within the Industry 4.0 Environment'. *EFNMS Euro Maintenance Conference*.
- [31] Ciocchetti, C. A. (2011) 'The eavesdropping employer: a twenty-first century framework for employee monitoring', *American Business Law Journal*, 48(2), pp. 285-369.
- [32] PWC (2018) *Anylogic* (Version 8.1.0). Available at: <https://www.anylogic.com/>.
- [33] da Silva, R. M., Benítez-Pina, I. F., Blos, M. F., Santos Filho, D. J. and Miyagi, P. E. (2015) 'Modeling of reconfigurable distributed manufacturing control systems', *IFAC-PapersOnLine*, 48(3), pp. 1284-1289.
- [34] Danielsson, O. (2016) 'Designing Augmented Reality Interfaces for Human-Robot Collaboration in Engine Assembly'.
- [35] Dennett, D. C. and Haugeland, J. (1987) 'Intentionality', *The Oxford companion to the mind*: Oxford University Press.
- [36] Ding, K., Chan, F. T., Zhang, X., Zhou, G. and Zhang, F. (2019) 'Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors', *International Journal of Production Research*, 57(20), pp. 6315-6334.
- [37] Dinges, D. F., Pack, F., Williams, K., Gillen, K. A., Powell, J. W., Ott, G. E., Aptowicz, C. and Pack, A. I. (1997) 'Cumulative sleepiness, mood disturbance, and psychomotor vigilance performance decrements during a week of sleep restricted to 4–5 hours per night', *Sleep*, 20(4), pp. 267-277.
-

-
- [38] Djuric, A. M., Urbanic, R. and Rickli, J. (2016) 'A Framework for Collaborative Robot (CoBot) Integration in Advanced Manufacturing Systems', *SAE International Journal of Materials and Manufacturing*, 9(2016-01-0337), pp. 457-464.
- [39] Driskell, J. E. and Salas, E. (2013) *Stress and human performance*. Psychology Press.
- [40] Dynamics, B. (2020a) *ATLAS*. Available at: <https://www.bostondynamics.com/atlas> (Accessed: 23/03/2020).
- [41] Dynamics, B. (2020b) *SPOT*: Boston Dynamics. Available at: <https://www.bostondynamics.com/spot> (Accessed: 23/03/2020).
- [42] Eckardt, M., Buchheim, A. and Gerngross, T. (2016) 'Investigation of an automated dry fiber preforming process for an aircraft fuselage demonstrator using collaborating robots', *CEAS Aeronautical Journal*, 7(3), pp. 429-440.
- [43] ElMaraghy, H. and ElMaraghy, W. (2016) 'Smart Adaptable Assembly Systems', *Procedia CIRP*, 44, pp. 4-13.
- [44] Enoka, R. M. and Duchateau, J. (2016) 'Translating fatigue to human performance', *Medicine and science in sports and exercise*, 48(11), pp. 2228.
- [45] Erol, S., Jäger, A., Hold, P., Ott, K. and Sihm, W. (2016) 'Tangible Industry 4.0: a scenario-based approach to learning for the future of production', *Procedia CIRP*, 54, pp. 13-18.
- [46] Evans, J. S. B. and Stanovich, K. E. (2013) 'Dual-process theories of higher cognition: Advancing the debate', *Perspectives on psychological science*, 8(3), pp. 223-241.
- [47] Fink, J. 'Anthropomorphism and human likeness in the design of robots and human-robot interaction'. *International Conference on Social Robotics*: Springer, 199-208.
- [48] Firoiu, V., Whitney, W. F. and Tenenbaum, J. B. (2017) 'Beating the World's Best at Super Smash Bros. with Deep Reinforcement Learning', *arXiv preprint arXiv:1702.06230*.
- [49] Forlizzi, J. and DiSalvo, C. 'Service robots in the domestic environment: a study of the roomba vacuum in the home'. *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*: ACM, 258-265.
- [50] Goldman, A. I. 2012. Theory of mind. Oxford Handbooks Online, <http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780195309799.001>.
- [51] Goldstone, J. A. (2015) 'Population ageing and global economic growth', *History & mathematics: Political demography & global ageing*, pp. 147-155.
- [52] Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep learning*. MIT Press.
- [53] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. 'Generative adversarial nets'. *Advances in neural information processing systems*, 2672-2680.
- [54] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G. and Cai, J. (2018) 'Recent advances in convolutional neural networks', *Pattern Recognition*, 77, pp. 354-377.
- [55] Guido, G., Rogano, D., Vitale, A., Astarita, V. and Festa, D. 'Big data for public transportation: A DSS framework'. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*: IEEE, 872-877.
- [56] Gurgun, L., Gunalp, O., Benazzouz, Y. and Gallissot, M. 'Self-aware cyber-physical systems and applications in smart buildings and cities'. *Proceedings of the Conference on Design, Automation and Test in Europe*: EDA Consortium, 1149-1154.
- [57] Güera, D. and Delp, E. J. 'Deepfake video detection using recurrent neural networks'. *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*: IEEE, 1-6.
- [58] Hallinan, B. and Striphas, T. (2016) 'Recommended for you: The Netflix Prize and the production of algorithmic culture', *New media & society*, 18(1), pp. 117-137.
-

-
- [59] Hancock, P., Williams, G. and Manning, C. (1995) 'Influence of task demand characteristics on workload and performance', *The International Journal of Aviation Psychology*, 5(1), pp. 63-86.
- [60] Hancock, P. A., Billings, D. R., Schaefer, K. E., Chen, J. Y., De Visser, E. J. and Parasuraman, R. (2011) 'A meta-analysis of factors affecting trust in human-robot interaction', *Human Factors*, 53(5), pp. 517-527.
- [61] Harari, Y. N. (2014) *Sapiens: A brief history of humankind*. Random House.
- [62] Hart, S. G. and Staveland, L. E. (1988) 'Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research', *Advances in psychology*: Elsevier, pp. 139-183.
- [63] Haykin, S., Fatemi, M., Setoodeh, P. and Xue, Y. (2012) 'Cognitive control', *Proceedings of the IEEE*, 100(12), pp. 3156-3169.
- [64] He, B. and Bai, K.-J. (2019) 'Digital twin-based sustainable intelligent manufacturing: a review', *Advances in Manufacturing*, pp. 1-21.
- [65] Heinisch, K. and Scheufele, R. (2017) 'Bottom-up or direct? Forecasting German GDP in a data-rich environment', *Empirical Economics*, pp. 1-41.
- [66] Heitmeyer, C. L. and Leonard, E. I. 'Obtaining trust in autonomous systems: tools for formal model synthesis and validation'. *Proceedings of the Third FME Workshop on Formal Methods in Software Engineering*: IEEE Press, 54-60.
- [67] Herron, C. and Braiden, P. (2007) 'Defining the foundation of lean manufacturing in the context of its origins (Japan)'.
[68] Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A. and Osband, I. 'Deep q-learning from demonstrations'. *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [69] Hilbert, M. (2016) 'Big data for development: A review of promises and challenges', *Development Policy Review*, 34(1), pp. 135-174.
- [70] Hoff, K. A. and Bashir, M. (2015) 'Trust in automation integrating empirical evidence on factors that influence trust', *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 57(3), pp. 407-434.
- [71] Hoffman, G. (2012) 'Embodied cognition for autonomous interactive robots', *Topics in cognitive science*, 4(4), pp. 759-772.
- [72] Hoffman, G. 'Evaluating fluency in human-robot collaboration'. *International conference on human-robot interaction (HRI), workshop on human robot collaboration*, 1-8.
- [73] Hsu, J. M. and Peters, S. C. 'Extending open dynamics engine for the DARPA virtual robotics challenge'. *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*: Springer, 37-48.
- [74] Isaak, J. and Hanna, M. J. (2018) 'User data privacy: Facebook, Cambridge Analytica, and privacy protection', *Computer*, 51(8), pp. 56-59.
- [75] Isla, D., Burke, R., Downie, M. and Blumberg, B. 'A layered brain architecture for synthetic creatures'. *International Joint Conference on Artificial Intelligence*: LAWRENCE ERLBAUM ASSOCIATES LTD, 1051-1058.
- [76] Jagadish, H., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R. and Shahabi, C. (2014) 'Big data and its technical challenges', *Communications of the ACM*, 57(7), pp. 86-94.
- [77] Kerkhof, G. A. (1985) 'Inter-individual differences in the human circadian system: a review', *Biological psychology*, 20(2), pp. 83-112.
- [78] Khalid, A., Kirisci, P., Ghrairi, Z., Thoben, K.-D. and Pannek, J. (2016) 'A methodology to develop collaborative robotic cyber physical systems for production environments', *Logistics Research*, 9(1), pp. 23.
-

-
- [79] Kirgis, F.-P., Katsos, P. and Kohlmaier, M. (2016) 'Collaborative robotics', *Robotic Fabrication in Architecture, Art and Design 2016*: Springer, pp. 448-453.
- [80] Kleinman, Z. (2016) 'Is your smartphone listening to you?', Available: BBC (Accessed 12/09/2019).
- [81] Koslowsky, M. and Babkoff, H. (1992) 'Meta-analysis of the relationship between total sleep deprivation and performance', *Chronobiology International*, 9(2), pp. 132-136.
- [82] Kotseruba, I., Gonzalez, O. J. A. and Tsotsos, J. K. (2016) 'A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications', *arXiv preprint arXiv:1610.08602*, pp. 1-74.
- [83] Kritzinger, W., Karner, M., Traar, G., Henjes, J. and Sihn, W. (2018) 'Digital Twin in manufacturing: A categorical literature review and classification', *IFAC-PapersOnLine*, 51(11), pp. 1016-1022.
- [84] Kusiak, A. (2019) 'Convolutional and generative adversarial neural networks in manufacturing', *International Journal of Production Research*, pp. 1-11.
- [85] Lab, M. M. (2020) *Moral Machine*. Available at: <http://moralmachine.mit.edu/>.
- [86] Laird, J. E. (2012) *The Soar cognitive architecture*. MIT press.
- [87] Laird, J. E., Newell, A. and Rosenbloom, P. S. (1987) 'Soar: An architecture for general intelligence', *Artificial intelligence*, 33(1), pp. 1-64.
- [88] LeCun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *Nature*, 521(7553), pp. 436-444.
- [89] Lee, E. A. 'Cyber physical systems: Design challenges'. *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*: IEEE, 363-369.
- [90] Lee, E. A. 'Cyber physical systems: Design challenges'. *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*: IEEE, 363-369.
- [91] Lee, J., Bagheri, B. and Jin, C. (2016) 'Introduction to cyber manufacturing', *Manufacturing Letters*, 8, pp. 11-15.
- [92] Lee, J., Bagheri, B. and Kao, H.-A. (2015) 'A cyber-physical systems architecture for industry 4.0-based manufacturing systems', *Manufacturing Letters*, 3, pp. 18-23.
- [93] Leitao, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T. and Colombo, A. W. (2016) 'Smart Agents in Industrial Cyber-Physical Systems', *Proceedings of the IEEE*, 104(5), pp. 1086-1101.
- [94] Leitao, P., Rodrigues, N., Barbosa, J., Turrin, C. and Pagani, A. (2015) 'Intelligent products: The grace experience', *Control Engineering Practice*, 42, pp. 95-105.
- [95] Leitão, P. (2009) 'Agent-based distributed manufacturing control: A state-of-the-art survey', *Engineering Applications of Artificial Intelligence*, 22(7), pp. 979-991.
- [96] Leitão, P. and Colombo, A. W. 'Petri net based methodology for the development of collaborative production systems'. *Emerging Technologies and Factory Automation, 2006. ETFA'06. IEEE Conference on*: IEEE, 819-826.
- [97] Leitão, P., Mařík, V. and Vrba, P. (2013) 'Past, present, and future of industrial agent applications', *IEEE Transactions on Industrial Informatics*, 9(4), pp. 2360-2372.
- [98] Leitão, P. and Restivo, F. (2006) 'ADACOR: A holonic architecture for agile and adaptive manufacturing control', *Computers in industry*, 57(2), pp. 121-130.
- [99] Leitão, P. and Strasser, T. 'Analyzing standardization needs for applying agent technology in industrial environments'. *Industrial Electronics (ISIE), 2016 IEEE 25th International Symposium on*: IEEE, 754-759.
- [100] Lemaignan, S., Warnier, M., Sisbot, E. A., Clodic, A. and Alami, R. (2017) 'Artificial cognition for social human-robot interaction: An implementation', *Artificial Intelligence*, 247, pp. 45-69.
-

-
- [101] Levy, K. E. (2015) 'The contexts of control: Information, power, and truck-driving work', *The Information Society*, 31(2), pp. 160-174.
- [102] Li, B.-h., Hou, B.-c., Yu, W.-t., Lu, X.-b. and Yang, C.-w. (2017) 'Applications of artificial intelligence in intelligent manufacturing: a review', *Frontiers of Information Technology & Electronic Engineering*, 18(1), pp. 86-96.
- [103] Lohani, M., Stokes, C., McCoy, M., Bailey, C. A. and Rivers, S. E. 'Social interaction moderates human-robot trust-reliance relationship and improves stress coping'. *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*: IEEE Press, 471-472.
- [104] Lorist, M. M., Kernell, D., Meijman, T. F. and Zijdewind, I. (2002) 'Motor fatigue and cognitive task performance in humans', *The Journal of physiology*, 545(1), pp. 313-319.
- [105] Lorist, M. M., Klein, M., Nieuwenhuis, S., De Jong, R., Mulder, G. and Meijman, T. F. (2000) 'Mental fatigue and task control: planning and preparation', *Psychophysiology*, 37(5), pp. 614-625.
- [106] Lowden, A., Moreno, C., Holmbäck, U., Lennernäs, M. and Tucker, P. (2010) 'Eating and shift work—effects on habits, metabolism, and performance', *Scandinavian journal of work, environment & health*, pp. 150-162.
- [107] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P. and Mordatch, I. (2017) 'Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments', *arXiv preprint arXiv:1706.02275*.
- [108] Marik, V. and McFarlane, D. (2005) 'Industrial adoption of agent-based technologies', *IEEE Intelligent Systems*, 20(1), pp. 27-35.
- [109] Marsland, S. (2015) *Machine learning: an algorithmic perspective*. CRC press.
- [110] Martin, K. and Freeman, R. E. (2003) 'Some problems with employee monitoring', *Journal of Business Ethics*, 43(4), pp. 353-361.
- [111] Matsas, E. and Vosniakos, G.-C. (2015) 'Design of a virtual reality training system for human-robot collaboration in manufacturing tasks', *International Journal on Interactive Design and Manufacturing (IJIDeM)*, pp. 1-15.
- [112] Matsugu, M., Mori, K., Mitari, Y. and Kaneda, Y. (2003) 'Subject independent facial expression recognition with robust face detection using a convolutional neural network', *Neural Networks*, 16(5), pp. 555-559.
- [113] Medsker, L. and Jain, L. (2001) 'Recurrent neural networks', *Design and Applications*, 5.
- [114] Mendes, J. M., Leitão, P., Restivo, F. and Colombo, A. W. 'Service-oriented agents for collaborative industrial automation and production systems'. *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*: Springer, 13-24.
- [115] Meskimen, J. (2019) *The Deeper Metrics of Christmas, Deep Fake Poem by Impressionist Jim Meskimen*. Available at: <https://www.youtube.com/watch?v=nD63QvIMB6k> (Accessed: 12/02/2019).
- [116] Miškuf, M. and Zolotová, I. 'Comparison between multi-class classifiers and deep learning with focus on industry 4.0'. *Cybernetics & Informatics (K&I), 2016*: IEEE, 1-5.
- [117] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M. (2013) 'Playing atari with deep reinforcement learning', *arXiv preprint arXiv:1312.5602*.
- [118] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K. and Ostrovski, G. (2015) 'Human-level control through deep reinforcement learning', *Nature*, 518(7540), pp. 529-533.
-

-
- [119] Modha, D. S., Ananthanarayanan, R., Esser, S. K., Ndirango, A., Sherbondy, A. J. and Singh, R. (2011) 'Cognitive computing', *Communications of the ACM*, 54(8), pp. 62-71.
- [120] Mohammed, A., Schmidt, B. and Wang, L. (2016) 'Active collision avoidance for human-robot collaboration driven by vision sensors', *International Journal of Computer Integrated Manufacturing*, pp. 1-11.
- [121] Moniz, A. B. and Krings, B.-J. (2016) 'Robots Working with Humans or Humans Working with Robots? Searching for Social Dimensions in New Human-Robot Interaction in Industry', *Societies*, 6(3), pp. 23.
- [122] Monostori, L. (2014) 'Cyber-physical production systems: Roots, expectations and R&D challenges', *Procedia CIRP*, 17, pp. 9-13.
- [123] Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W. and Ueda, K. (2016) 'Cyber-physical systems in manufacturing', *CIRP Annals-Manufacturing Technology*, 65(2), pp. 621-641.
- [124] Mori, M. 1970. Bukimi no tani [The uncanny valley], *Energy*, [Online] 7 (4), 33-35.
- [125] Musil, A., Musil, J., Weyns, D., Bures, T., Muccini, H. and Sharaf, M. (2017) 'Patterns for Self-Adaptation in Cyber-Physical Systems', *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*: Springer, pp. 331-368.
- [126] Nichols, S. 2018. Your Phone Is Listening and it's Not Paranoia. *Vice*.
- [127] Noor, A. K. (2015) 'Potential of cognitive computing and cognitive systems', *Open Engineering*, 5(1), pp. 75-88.
- [128] Paelke, V. 'Augmented reality in the smart factory: Supporting workers in an industry 4.0. environment'. *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*: IEEE, 1-4.
- [129] Pasqualetti, F., Dörfler, F. and Bullo, F. (2013) 'Attack detection and identification in cyber-physical systems', *IEEE Transactions on Automatic Control*, 58(11), pp. 2715-2729.
- [130] Permin, E., Bertelsmeier, F., Blum, M., Bützler, J., Haag, S., Kuz, S., Özdemir, D., Stemmler, S., Thombansen, U. and Schmitt, R. (2016) 'Self-optimizing production systems', *Procedia CIRP*, 41, pp. 417-422.
- [131] Pipattanasomporn, M., Feroze, H. and Rahman, S. 'Multi-agent systems in a distributed smart grid: Design and implementation'. *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*: IEEE, 1-8.
- [132] Pirvu, B.-C., Zamfirescu, C.-B. and Gorecky, D. (2016) 'Engineering insights from an anthropocentric cyber-physical system: A case study for an assembly station', *Mechatronics*, 34, pp. 147-159.
- [133] Qiu, J., Wu, Q., Ding, G., Xu, Y. and Feng, S. (2016) 'A survey of machine learning for big data processing', *EURASIP Journal on Advances in Signal Processing*, 2016(1), pp. 67.
- [134] Raibert, M., Blankespoor, K., Nelson, G. and Playter, R. (2008) 'Bigdog, the rough-terrain quadruped robot', *IFAC Proceedings Volumes*, 41(2), pp. 10822-10825.
- [135] Rodrigues, N., Pereira, A. and Leitão, P. 'Adaptive multi-agent system for a washing machine production line'. *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*: Springer, 212-223.
- [136] Romero, D., Bernus, P., Noran, O., Stahre, J. and Fast-Berglund, Å. 'The Operator 4.0: Human Cyber-Physical Systems & Adaptive Automation towards Human-Automation Symbiosis Work Systems'. *APMS (Advances in Production Management Systems)*.
- [137] Rozo, L., Calinon, S., Caldwell, D. G., Jimenez, P. and Torras, C. (2016) 'Learning physical collaborative robot behaviors from human demonstrations', *IEEE Transactions on Robotics*, 32(3), pp. 513-527.
-

-
- [138] Rusinko, C. (2007) 'Green manufacturing: an evaluation of environmentally sustainable manufacturing practices and their impact on competitive outcomes', *IEEE Transactions on Engineering Management*, 54(3), pp. 445-454.
- [139] Sachdev, S. (2017) *Would you push the fat man off the bridge?* Available at: <http://shikharsachdev.com/trolley-problem/> (Accessed: 03/03/17 2017).
- [140] Sadrfaridpour, B., Saeidi, H., Burke, J., Madathil, K. and Wang, Y. (2016) 'Modeling and control of trust in human-robot collaborative manufacturing', *Robust Intelligence and Trust in Autonomous Systems*: Springer, pp. 115-141.
- [141] Sak, H., Senior, A. and Beaufays, F. 'Long short-term memory recurrent neural network architectures for large scale acoustic modeling'. *Fifteenth Annual Conference of the International Speech Communication Association*.
- [142] Salvucci, D., Boer, E. and Liu, A. (2001) 'Toward an integrated model of driver behavior in cognitive architecture', *Transportation Research Record: Journal of the Transportation Research Board*, (1779), pp. 9-16.
- [143] Schmidhuber, J. (2015) 'Deep learning in neural networks: An overview', *Neural networks*, 61, pp. 85-117.
- [144] Scholze, S. and Barata, J. 'Context awareness for flexible manufacturing systems using cyber physical approaches'. *Doctoral Conference on Computing, Electrical and Industrial Systems*: Springer, 107-115.
- [145] Schuh, G., Potente, T., Wesch-Potente, C., Weber, A. R. and Prote, J.-P. (2014) 'Collaboration Mechanisms to increase Productivity in the Context of Industrie 4.0', *Procedia CIRP*, 19, pp. 51-56.
- [146] Schönsleben, P., Fantana, F. and Duchi, A. 'What benefits do initiatives such as Industry 4.0 offer for production locations in high-wage countries?', *CIRP 50th Conference on Manufacturing Systems*, Taichung, Taiwan: Elsevier.
- [147] Shafiq, S. I., Sanin, C., Szczerbicki, E. and Toro, C. (2016) 'Virtual engineering factory: Creating experience base for industry 4.0', *Cybernetics and Systems*, 47(1-2), pp. 32-47.
- [148] Sharp, M., Ak, R. and Hedberg Jr, T. (2018) 'A survey of the advancing use and development of machine learning in smart manufacturing', *Journal of Manufacturing Systems*.
- [149] Shaw, S.-L., Tsou, M.-H. and Ye, X. (2016) 'Human dynamics in the mobile and big data era', *International Journal of Geographical Information Science*, 30(9), pp. 1687-1693.
- [150] Shi, D., Fan, W., Xiao, Y., Lin, T. and Xing, C. (2020) 'Intelligent scheduling of discrete automated production line via deep reinforcement learning', *International Journal of Production Research*, pp. 1-19.
- [151] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V. and Lanctot, M. (2016) 'Mastering the game of Go with deep neural networks and tree search', *Nature*, 529(7587), pp. 484-489.
- [152] Spezzano, G. and Vinci, A. (2015) 'Pattern detection in cyber-physical systems', *Procedia Computer Science*, 52, pp. 1016-1021.
- [153] *Jurassic Park*, 1993. Directed by Spielberg, S. USA: Universal Pictures.
- [154] Srnicek, N. (2017) *Platform capitalism*. John Wiley & Sons.
- [155] Sutton, R. S. (1985) 'Temporal Credit Assignment in Reinforcement Learning'.
- [156] Sutton, R. S. and Barto, A. G. (1998) *Introduction to reinforcement learning*. MIT press Cambridge.
-

-
- [158] Sutton, R. S. and Barto, A. G. (2018) *Reinforcement learning: An introduction*. MIT press.
- [159] Tao, F., Zhang, L., Venkatesh, V., Luo, Y. and Cheng, Y. (2011) 'Cloud manufacturing: a computing and service-oriented manufacturing model', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 225(10), pp. 1969-1976.
- [160] Taylor, F. W. (1911) 'The principles of scientific management New York', NY: *Harper & Brothers*.
- [161] Taylor, F. W. (2004) *Scientific management*. Routledge.
- [162] Author (2018) *Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0*. (Version 1.0.0-beta_3). Available at: <http://deeplearning4j.org>.
- [163] Teiwes, J., Bänziger, T., Kunz, A. and Wegener, K. 'Identifying the potential of human-robot collaboration in automotive assembly lines using a standardised work description'. *Automation and Computing (ICAC), 2016 22nd International Conference on: IEEE*, 78-83.
- [164] Testu, F. and Clarisse, R. (1999) 'Time-of-day and day-of-week effects on mnemonic performance', *Chronobiology international*, 16(4), pp. 491-503.
- [165] Turing, A. M. (1950) 'Computing machinery and intelligence', *Mind*, 59(236), pp. 433-460.
- [166] Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P. (1998) 'Reference architecture for holonic manufacturing systems: PROSA', *Computers in industry*, 37(3), pp. 255-274.
- [167] van Otterlo, M. and Wiering, M. (2012) 'Reinforcement learning and markov decision processes', *Reinforcement Learning: Springer*, pp. 3-42.
- [168] Vernon, D. (2014) *Artificial cognitive systems: A primer*. MIT Press.
- [169] Vorotnikov, S., Ermishin, K., Nazarova, A. and Yuschenko, A. 'Multi-agent robotic systems in collaborative robotics'. *International Conference on Interactive Collaborative Robotics: Springer*, 270-279.
- [170] Vox 2017. <yt-formatted-string class="style-scope ytd-video-primary-info-renderer" force-default-style="" style="word-break: break-word;">How job surveillance is changing trucking in America</yt-formatted-string>. *Shift Change*. Youtube.
- [171] Vyatkin, V. (2013) 'Software engineering in industrial automation: State-of-the-art review', *IEEE Transactions on Industrial Informatics*, 9(3), pp. 1234-1249.
- [172] Wang, F.-Y. (2005) 'Agent-based control for networked traffic management systems', *IEEE Intelligent Systems*, 20(5), pp. 92-96.
- [173] Wang, L., Törngren, M. and Onori, M. (2015) 'Current status and advancement of cyber-physical systems in manufacturing', *Journal of Manufacturing Systems*, 37(Part 2), pp. 517-527.
- [174] Wang, Y. and Zhang, F. 2017. Trends in Control and Decision-Making for Human-Robot Collaboration Systems. Springer.
- [175] Warta, S. F., Kapalo, K. A., Best, A. and Fiore, S. M. 'Similarity, Complementarity, and Agency in HRI: Theoretical Issues in Shifting the Perception of Robots from Tools to Teammates'. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting: SAGE Publications Sage CA: Los Angeles, CA*, 1230-1234.
- [176] Watkins, C. J. and Dayan, P. (1992) 'Q-learning', *Machine learning*, 8(3-4), pp. 279-292.
- [177] Weyer, S., Schmitt, M., Ohmer, M. and Gorecky, D. (2015) 'Towards Industry 4.0-Standardization as the crucial challenge for highly modular, multi-vendor production systems', *IFAC-PapersOnLine*, 48(3), pp. 579-584.
-

-
- [178] Wiltshire, T. J., Barber, D. and Fiore, S. M. 'Towards modeling social-cognitive mechanisms in robots to facilitate human-robot teaming'. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*: SAGE Publications Sage CA: Los Angeles, CA, 1278-1282.
- [179] Wiltshire, T. J., Snow, S. L., Lobato, E. J. and Fiore, S. M. 'Leveraging social judgment theory to examine the relationship between social cues and signals in human-robot interactions'. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*: SAGE Publications Sage CA: Los Angeles, CA, 1336-1340.
- [180] Wiltshire, T. J., Warta, S. F., Barber, D. and Fiore, S. M. (2016) 'Enabling robotic social intelligence by engineering human social-cognitive mechanisms', *Cognitive Systems Research*.
- [181] Witten, I. H., Frank, E., Hall, M. A. and Pal, C. J. (2016) *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.
- [182] Wooldridge, M. and Jennings, N. R. (1995) 'Intelligent agents: Theory and practice', *The knowledge engineering review*, 10(02), pp. 115-152.
- [183] Wuest, T., Irgens, C. and Thoben, K.-D. (2014) 'An approach to monitoring quality in manufacturing using supervised machine learning on product state data', *Journal of Intelligent Manufacturing*, 25(5), pp. 1167-1180.
- [184] Young, J. E., Sung, J., Volda, A., Sharlin, E., Igarashi, T., Christensen, H. I. and Grinter, R. E. (2011) 'Evaluating human-robot interaction', *International Journal of Social Robotics*, 3(1), pp. 53-67.
- [185] Zhang, Y., Qian, C., Lv, J. and Liu, Y. (2016) 'Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor', *IEEE Transactions on Industrial Informatics*.
- [186] Zhong, R. Y., Xu, C., Chen, C. and Huang, G. Q. (2017) 'Big data analytics for physical internet-based intelligent manufacturing shop floors', *International journal of production research*, 55(9), pp. 2610-2621.
- [187] Zhou, K., Liu, T. and Zhou, L. 'Industry 4.0: Towards future industrial opportunities and challenges'. *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on*: IEEE, 2147-2152.
- [188] Zuboff, S. (2015) 'Big other: surveillance capitalism and the prospects of an information civilization', *Journal of Information Technology*, 30(1), pp. 75-89.
- [189] Zuboff, S. (2019) *The age of surveillance capitalism: The fight for a human future at the new frontier of power*. Profile Books.
- [190] Zuehlke, D. (2010) 'SmartFactory—Towards a factory-of-things', *Annual Reviews in Control*, 34(1), pp. 129-138.

APPENDICIES

Appendix.1. Java code of the developed *Software Agent*

```
package org.cardiffuniversity.c1225978;

import au.com.bytecode.opencsv.CSVReader;
import au.com.bytecode.opencsv.CSVWriter;
import org.deeplearning4j.nn.api.OptimizationAlgorithm;
import org.deeplearning4j.nn.conf.MultiLayerConfiguration;
import org.deeplearning4j.nn.conf.NeuralNetConfiguration;
import org.deeplearning4j.nn.conf.layers.DenseLayer;
import org.deeplearning4j.nn.conf.layers.OutputLayer;
import org.deeplearning4j.nn.multilayer.MultiLayerNetwork;
import org.deeplearning4j.nn.weights.WeightInit;
import org.deeplearning4j.optimize.listeners.ScoreIterationListener;
import org.deeplearning4j.rl4j.learning.Learning;
import org.deeplearning4j.util.ModelSerializer;
import org.nd4j.linalg.activations.Activation;
import org.nd4j.linalg.api.ndarray.INDArray;
import org.nd4j.linalg.factory.Nd4j;
import org.nd4j.linalg.learning.config.Adam;
import org.nd4j.linalg.learning.config.RmsProp;
import org.nd4j.linalg.lossfunctions.LossFunctions;
import java.io.*;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ThreadLocalRandom;

//New instance of class loaded on Anylogic simulation start
//(Possible multiple instances for multiple operators)
//Default constructor generates new DQN
//Param constructor loads previously serialized DQN
public class SoftwareAgent{

    //VARIABLES
    private INDArray Observation; //CurrentState
    private INDArray Scores;
    private String Action = "NONE"; //CurrentAction
    private int action = 0; //placeholder int action
    private double lastReward;
    private double cumulativeReward = 0;
    private double productCount = 0;
    private double cumulativeIdle = 0;
    private double OperatorNumber = 1;
    private int iteration = 0;
    private String filename = "newAgent"; //save directory
    private File metadataFile;
    private File outputFile;
    private CSVWriter writer;
    protected MultiLayerNetwork DQN;
    protected MultiLayerNetwork DQN_TARGET;

    private AnyLogicActionSpace actionSpace = new
AnyLogicActionSpace("INCREASESPEED", "NEXT ACTION", "DECREASESPEED");
    private AnyLogicObservationSpace observationSpace = new
AnyLogicObservationSpace(
        "Current CT",
        "Calculated Target CT",
        "Collaborative Target CT",
        "Speed Factor",
        "Idle Time",
        "Idle Time CT");
```

```

//Hyperparameters
private double gamma = 0.7;
private double epsilon = 1;
private double epsilon_min = 0.01;
private double epsilon_decay = 0.99;
boolean LRdecay = false;
double LRrate = 0.995;
private int memorySize = 10000;
private int batchSize = 100;
private int numHiddenNodes = 5;

//Memory
//CircularFifoQueue<Transition<String>> memory = new
CircularFifoQueue<Transition<String>>(2500);
protected ExpReplay memory = new ExpReplay(memorySize, batchSize,
this.observationSpace.getSize());

//Methods Declarations
//CONSTRUCTORS

//Default Constructor
public SoftwareAgent() {

    boolean init = false;

    //attempt to load memory
    //Determines whether to generate new agent
    try {
        //Load Model
        this.DQN =
ModelSerializer.restoreMultiLayerNetwork(filename+".zip",true);
        System.out.println("Model Loaded");

        //Load ExpReplay
        FileInputStream fin= new FileInputStream (filename+".tmp");
        ObjectInputStream ois = new ObjectInputStream(fin);
        this.memory = (ExpReplay) ois.readObject();
        fin.close();
        System.out.println("Memory Loaded");
        //Set Normalization
        this.memory.Normalize = true;
    } catch (IOException e) {
        System.out.println("Load Error, Check Pathname");
        System.out.println();
        init = true;
    } catch (ClassNotFoundException c){
        System.out.println("Memory Load Error");
        init = true;
        System.out.println();
    }
    if(init) {
        //Define DQN Configuration
        MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
            .seed(420)
            .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
            .updater(new Adam(0.0025))
            .weightInit(WeightInit.XAVIER)
            .list()
            .layer(0, new DenseLayer.Builder()
                .nIn(observationSpace.getSize())
                .nOut(numHiddenNodes)
                .activation(Activation.RELU)
                .build())
            .layer(1, new OutputLayer.Builder()
                .nIn(numHiddenNodes)

```

```

        .nout(actionSpace.getSize())
        .activation(Activation.IDENTITY)
        .lossFunction(LossFunctions.LossFunction.MSE)
        .build()
    .build();

    //Define DQN from configuration
    MultiLayerNetwork model = new MultiLayerNetwork(conf);
    model.init();
    DQN = model.clone();

    //Set Target DQN
    DQN_TARGET = model.clone();

    //Setup data Collection
    this.iteration=1;
    String dir = "D:/"+filename+"_data/";
    File directory = new File(dir);
    boolean bool = directory.mkdirs();
    System.out.println(dir);
    this.metadataFile = new File(dir+"metadata.csv");
    this.outputFile = new File(dir+String.valueOf(this.iteration)+".csv");
    try {
        FileWriter metadatafile = new FileWriter(this.metadataFile);
        CSVWriter metadatawriter = new CSVWriter(metadatafile);
        System.out.println("Metadata File defined");
        //add header
        String[] header = {"Iteration",
            "Cumulative Reward",
            "Product Count",
            "Total Idle Time",
            "Productivity"};
        metadatawriter.writeNext(header);
    }
    catch(IOException e){
        System.out.println("Metatdata File creation failed...");
    }
    //create data file for iteration
    createDataFile();

}else{
    //Setup data Collection
    this.metadataFile = new File("D:/"+filename+"_data/metadata.csv");
    try {
        FileReader inFile = new FileReader(this.metadataFile);
        CSVReader csvinFile = new CSVReader(inFile);
        List<String[]> list =csvinFile.readAll();
        String[] lastiter = list.get(list.size()-1);
        this.iteration = Integer.parseInt(lastiter[0]);
        this.iteration++;
        inFile.close();
    }catch(IOException e) {
        System.out.println("Metadata Update Failed...");
    }
    //create data file for iteration
    this.outputFile = new
File("D:/"+filename+"_data/"+String.valueOf(iteration)+".csv");
    createDataFile();

    //Set Target DQN
    this.DQN_TARGET = this.DQN.clone();

    //Implement Learning Rate Decay
    if(LRdecay){
        double lr = DQN.getLearningRate(0);
        this.DQN.setLearningRate(lr*this.LRrate);

```

```

    }

    //Implement epsilon decay
    if(this.epsilon>this.epsilon_min) {
        for(int j=0;j<this.iteration;j++){
            this.epsilon = this.epsilon*this.epsilon_decay;
        }
    }
}

//ACT
//Returns the action with highest expected reward
public int GetMaxAction(INDArray input) {
    this.Observation = input;
    double k = Math.random();
    if (k < this.epsilon){
        int action = this.actionSpace.randomAction();

        //Sets action and current State on exit.
        this.Observation = input.dup();
        this.Scores = Nd4j.zeros(actionSpace.getSize());
        this.action = action;
        this.Action = this.actionSpace.encode(action);
        return action;
    }else{
        //Option for normalized inputs:
        if(this.memory.Normalize = true) {
            INDArray ninput = (this.memory.NormalizeArray(input));
            INDArray output = this.DQN.output(ninput);
        }
        else{
            INDArray output = this.DQN.output(input);
        }
        this.Scores = output.dup();
        int action = Learning.getMaxAction(output);

        //Sets action and current State on exit.
        this.Observation = input.dup();
        this.action = action;
        this.Action = this.actionSpace.encode(action);
        return action;
    }
}

//REMEMBER
//Store transition into memory fifo queue
public void StepReply(INDArray newstate, double reward) {
    if(this.Action.equals("NONE")){
        System.out.println("Invalid Action Pair");
        System.out.println("Action: "+ Action);
    }else {
        //Create New Transition and store values
        Transition<String> trans = new Transition<>();
        trans.setObservation(Observation);
        trans.setAction(Action);
        trans.setReward(reward);
        trans.setTerminal(false);
        trans.setNextObservation(newstate);
        this.memory.store(trans);
        this.lastReward = reward;
        this.cumulativeReward+=reward;
        trainstep(trans);
    }
}

```

```

    }
}

//REPLAY
//Training step on single transition
public void trainstep(Transition<String> step){
    //Get values from transition
    INDArray obs = step.getObservation();
    String action = step.getAction();
    double rew = step.getReward();
    INDArray nextobs = step.getNextObservation();

    //System.out.println("obs: " + obs);
    //System.out.println("nextobs:"+ nextobs);

    //get future reward estimates as array from current state
    //get rewards from next state
    //next state reward for max action taken at state multiplied by gamma
    //above value supplants prediction from state in array
    //this INDArray used to train as label
    INDArray target = this.DQN_TARGET.output(obs);
    INDArray target_future = this.DQN_TARGET.output(nextobs);

    //Get max reward at s+1
    INDArray tempQIndex = target_future.argmax();
    int tempQ = tempQIndex.getInt(0, 0);
    double q = target_future.getDouble(0, tempQ);

    //Calculate sum of discounted sum of future rewards
    rew += (gamma * q);

    //Calculate index of max action at state
    int tempIndex = this.actionSpace.decode(Action);

    //Adjust value from observation
    target.putScalar(tempIndex, rew);

    //Network trained on state and rewards array
    this.DQN.fit(obs, target);
}

//Trains network on batch of transitions
//gets batch from memory and iterates training step over batch
public void trainbatch() {
    long timestamp = System.currentTimeMillis();
    if (this.memory.size() < batchsize) {
        System.out.println("Not Enough Transitions");
        System.out.println();
    } else {
        System.out.println("Training Step:");
        for (int j = 0; j < batchsize; j++) {
            int k =
ThreadLocalRandom.current().nextInt(0,this.memory.memorysize-1);
            Transition<String> trans = this.memory.get(k);
            trainstep(trans);
        }
    }
    long t2 = System.currentTimeMillis()-timestamp;
    System.out.println("Time to train batch: "+t2+"ms");
}

//Update target DQN
public void UpdateTargets() {
    this.DQN_TARGET = this.DQN.clone();
}
}

```

```

//Train network on whole episode at end.
public void trainEpisode(){
    //Get batch equal to memory size
    ArrayList<Transition<String>> ShuffledMemory =
memory.getNewestBatch(memorySize);

    for (int j = 0; j < ShuffledMemory.size(); j++) {

        INDArray obs = ShuffledMemory.get(j).getObservation();
        String action = ShuffledMemory.get(j).getAction();
        double rew = ShuffledMemory.get(j).getReward();
        INDArray nextobs = ShuffledMemory.get(j).getNextObservation();

        //get future reward estimates as array from current state
        //get rewards from next state
        //next state reward for max action taken at state multiplied by gamma
        //above value supplants prediction from state in array
        //this INDArray used to train as label
        INDArray target = this.DQN_TARGET.output(obs);
        INDArray target_future = this.DQN_TARGET.output(nextobs);

        INDArray tempQIndex = target_future.argmax();
        int tempQ = tempQIndex.getInt(0, 0);
        double q = 0;
        q += target_future.getDouble(0, tempQ);
        rew += (gamma * q);

        //Calculate index of max action at state
        int tempIndex = this.actionSpace.decode(action);
        target.putScalar(tempIndex, rew);

        //Network trained on state and rewards array
        this.DQN.fit(obs, target);
    }
}

//UTILITIES -----
//decode action from Int
public String decode(int action){
    String newAction = this.actionSpace.encode(action);
    return newAction;
}

//Set Learning Rate
public void setLearningRate(double lr){
    this.DQN.setLearningRate(lr);
    this.DQN_TARGET = this.DQN.clone();
}

//Increases Product Count by 1
public void iterateProductCount(){
    this.productCount++;
}

//Takes a reward and accumates with Cumulative Idle variable
public void iterateCumulativeIdle(double idlestep){
    this.cumulativeIdle+=idlestep;
}

//Store data Instance
public void storeData(){
    int i = 0;
    String[] data = new
String[this.observationSpace.getSize()+this.actionSpace.getSize()+3];

```

```

while(i<this.observationSpace.getSize()){
    data[i] = Double.toString(this.Observation.getDouble(i));
    i++;
}
for (int j=0; j<this.actionSpace.getSize(); j++) {
    data[i] = Double.toString(this.Scores.getDouble(j));
    i++;
}
data[i] = this.Action;
i++;
data[i] = Double.toString(this.lastReward);
i++;
data[i] = Double.toString(this.OperatorNumber);
try {
    //add data
    this.writer.writeNext(data);
}
catch(Exception e){
    System.out.println("Data Write Failed...");
}
}

//Store MetaData for Iteration
public void storeMetaData(){
    int i = 0;
    String[] data = new String[5];
    data[i] = String.valueOf(this.iteration);
    i++;
    data[i] = String.valueOf(this.cumulativeReward);
    i++;
    data[i] = String.valueOf(this.productCount);
    i++;
    data[i] = String.valueOf(this.cumulativeIdle);
    i++;
    double productivity = ((this.cumulativeIdle)/this.productCount);
    data[i] = String.valueOf(productivity);
    try {
        FileReader metadatafilein = new FileReader(this.metadataFile);
        CSVReader metadatareader = new CSVReader(metadatafilein);
        List<String[]> list = metadatareader.readAll();
        list.add(data);
        FileWriter metadatafile = new FileWriter(this.metadataFile);
        CSVWriter metadatawriter = new CSVWriter(metadatafile);
        //add data
        metadatawriter.writeAll(list);
        metadatawriter.close();
        metadatafile.close();
        this.writer.close();
    }
    catch(Exception e){
        System.out.println("Data Write Failed...");
    }
}

//Save Network
public void SaveNetwork(String filename, String memFilename) {
    try {
        //Save the model
        File locationToSave = new File(filename); //Where to save the
network. Note: the file is in .zip format - can be opened externally
        boolean saveUpdater = true;
//Updater: i.e., the state for Momentum, RMSProp, Adagrad etc.
        ModelSerializer.writeModel(this.DQN, locationToSave, saveUpdater);
        System.out.println("Model Saved");

        //Save the Exp Replay

```

```

        FileOutputStream fout = new FileOutputStream (memFilename);
        ObjectOutputStream oos = new ObjectOutputStream(fout);
        oos.writeObject(this.memory);
        fout.close();
        System.out.println("Memory Saved");

    } catch (IOException e) {
        System.out.println("Network Save Error, Check Pathname");
        System.out.println();
    }
}

public void SaveNetwork() {
    try {
        //Save the model
        File locationToSave = new File(filename+".zip");           //Save Location
        boolean saveUpdater = true;                               //Save Updater
        ModelSerializer.writeModel(this.DQN, locationToSave, saveUpdater);
        System.out.println("Model Saved");

        //Save the Exp Replay
        FileOutputStream fout = new FileOutputStream (filename+".tmp");
        ObjectOutputStream oos = new ObjectOutputStream(fout);
        oos.writeObject(this.memory);
        fout.close();
        System.out.println("Memory Saved");

    } catch (IOException e) {
        System.out.println("Network Save Error, Check Pathname");
        System.out.println();
    }
}

//Destructor
public void destroyAgent(){
    //some actions to be taken on destruction of agent object
    //close files.
    storeMetaData();
}

//Creates .csv file recording data for each iteration
private void createDataFile(){
    try {
        FileWriter outputfileWriter = new FileWriter(this.outputFile);
        this.writer = new CSVWriter(outputfileWriter);
        //add header
        String[] header = {
            "Current CT",
            "Calculated Target CT",
            "Collaborative Target CT",
            "Speed Factor",
            "Idle Time",
            "Collaborative Idle",
            "Score 1",
            "Score 2",
            "Score 3",
            "Action Chosen",
            "Reward",
            "OperatorNumber"};
        this.writer.writeNext(header);
    }
    catch(IOException e){
        System.out.println("Iteration datafile creation failed...");
    }
}

//Reset Memory

```

```

    public void resetMemory(){
        this.memory = new ExpReplay(memorySize, batchsize,
this.observationSpace.getSize());
    }

    //PrintMemory
    public void printMemory(){
        System.out.println("Training Batch:");
        for(int i = 0; i<memory.size();i++){
            memory.get(i).print();
        }
        System.out.println();
    }

    //Getter for last action
    public int getLastAction(){
        int ac = action;
        return ac;
    }

    //Covert action to string
    public String actionToString(int action){
        String Action = this.actionSpace.encode(action);
        return Action;
    }

    //SETTERS
    //Observation Setter
    public void setObservation(INDArray obs){
        this.Observation = obs;
    }
    //Action Setter
    public void setAction(int action){
        this.action = action;
        this.Action = actionSpace.encode(action);
    }

    //ActionSpace setter
    public void setActionSpace(AnyLogicActionSpace as){
        this.actionSpace = as;
    }
    //Epsilon Setter
    public void setEpsilon(double value){
        this.epsilon = value;
    }
    public double getEpsilon(){
        return this.epsilon;
    }
    public int getIteration() {
        return this.iteration;
    }

    public void setProductCount(double value){
        this.productCount = value;
    }

    //Operator Number Setter
    public void setOperatorNumber(int on){
        this.OperatorNumber = on;
    };

    //End of Class Def
}

```
