B.Tech Project Report

On

A Social Network based Platform to Implement Collaborative Learning and Distributed Decision Making for Industrial Assets

By

Maharshi Dhada (140003015)

DISCIPLINE OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2017

A Social Network based Platform to Implement Collaborative Learning and Distributed Decision Making for Industrial Assets

---

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree

of

Bachelor of Technology

in

MECHANICAL ENGINEERING

Submitted by:

Maharshi Dhada (140003015)

Guided by:

Dr. Bhupesh Kumar Lad

Associate Professor, IIT Indore

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2017

CANDIDATE'S DECLARATION

I hereby declare that the project entitled **"A Social Network based Platform to Implement Collaborative Learning and Distributed Decision Making for Industrial Assets"** submitted in partial fulfilment for the award of the degree of Bachelor of Technology in 'Mechanical Engineering' completed under the supervision of Dr. Bhupesh Kumar Lad, Associate Professor, Discipline of Mechanical Engineering, IIT Indore is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Maharshi Dhada
B. Tech., Mechanical Engineering
140003015

CERTIFICATE BY BTP GUIDE

It is certified that the above statement made by the student is correct to the best of my knowledge.

Dr. Bhupesh Kumar Lad
Associate Professor
IIT Indore

Foreword

This report on **"A Social Network based Platform to Implement Collaborative Learning and Distributed Decision Making for Industrial Assets"** is prepared under the guidance of Dr. Bhupesh Kumar Lad.

Through this report I have tried to give a detailed description of the work done under the project titled 'Smart Manufacturing', by me and my partner Kshitij Bakliwal (140002018, Discipline of Electrical Engineering).

I have tried to the best of my abilities and knowledge to explain the contents in a lucid manner. I have also added images and figures depicting the effect of the techniques used to make the report more illustrative.

Maharshi Dhada
B. Tech., IV Year
Discipline of Mechanical Engineering
IIT Indore

Acknowledgements

I would like to express my deep gratitude to Dr. Ajith Kumar Parlikad (Senior Lecturer, University of Cambridge), Dr. Abhishek Srivastava (Associate Professor, IIT Indore), Mr. Amit Kumar Jain (PhD Scholar, Industrial Research Lab, IIT Indore) and Mr. Adrià Salvador Palau (PhD Candidate, University of Cambridge) for their constant support and valuable insights to help us with various challenges we faced in the course of this research project.

I also thank the staff and other researchers in the Smart Manufacturing Lab, IIT Indore and at the Institute for Manufacturing, University of Cambridge who were always there to help us unconditionally. This project would not have been successfully completed without their aid.

I acknowledge the Royal Academy of Engineering, UK for generously funding our project under the Newton-Bhabha Scheme, the visit to Cambridge (UK) and prompt support whenever we needed it.

Finally, I am immensely thankful to **Dr. Bhupesh K. Lad**, for his continuous guidance and for being such a motivating mentor, and I certainly acknowledge the facilities and resources provided to us by the Indian Institute of Technology Indore. This project has been completed by me in partnership with my colleague, **Kshitij Bakliwal**, B. Tech. IV Year, Discipline of Electrical Engineering (140002018).

Maharshi Dhada
B. Tech., IV Year
Discipline of Mechanical Engineering
IIT Indore

Abstract

Employing latest technologies from industrial paradigms such as Smart Manufacturing, a Multi-Agent System architecture has been designed and developed in this project. The architecture has further been implemented on three distinct industrial systems, each employing different algorithms and objectives. Relevant literature was thoroughly reviewed during the course of this project to gain newer perspectives and keep in pace with modern trends in the Manufacturing sector. The promising results obtained from the implementation of our architecture and algorithms serve as evidence of the applicability of our work for the real-world industries.

This report aims to give a detailed explanation of the motivation for the objective of the project, research done, challenges faced, novelty of the algorithms developed and the impressive results obtained.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Social Internet of Things

Accuracy, precision and cost-effectiveness of sensing technologies have improved in recent years, leading to extensive instrumentation of industrial assets and emergence of Big Data. Computers have become smart, compact, powerful and capable of operating over cloud servers. With advancement in communication technologies, low-cost transfer of significant amount of data over the internet is now possible [Ganchev et al., 2016]. Integrating all these developments with the objects of everyday use has led to a network of connected objects called the 'Internet of Things' (IoT). [Li et al., 2015]

Thanks to these technological advancements, up to date industrial assets are able to generate extensive data that reflects the system performance. Also, technologies such as RFID, smart cards, embedded systems, Wi-Fi, and Bluetooth communication have enabled automatised machine to machine communications to take place [Unland, 2015]. Such technologies make it possible to harness the benefits of the data generated in industrial environments. This extension of the notion of IoT is termed as the 'Industrial Internet of Things' (IIoT) [Evans and Annunziata, 2012, Xu et al., 2014].

In the IIoT, each asset has a Digital Twin, which is its cyber model/ replica, containing the asset data acquired from various sources. This Digital Twin is part of a network of several other twins with their corresponding assets, together forming a network of industrial assets. It is the Digital Twins where the local analytics are implemented over the asset-data. The streams of data flowing into the twins are analysed by human experts or computational algorithms to get a perception about the surroundings, performance, and the health conditions of the assets [GE Digital, 2017]. Embedded computers monitor and control the physical processes, usually with feedback loops. This way, physical processes shape the computations and vice-versa [Lee, 2008]. The integration of computation with physical processes described above forms a Cyber-Physical System (CPS) [Jeschke et al., 2017].

Further benefits can be harnessed by integrating human-like social networking capabilities into IoT. This notion of SIoT (Social Internet of Things), has been gaining momentum in the areas of product life-cycle management, traffic routing, and workplace help and support. The integration of social networks and IoT can be extended to improve system-level performance in an asset fleet. In this project, a quantitative method to identify groups of similar assets, or 'friends' is proposed. Subsequently, condition data, and the diagnostics or the prognostics knowledge shared among these assets can improve the accuracy of the asset's computations [Li et al., 2018]. Several projects like the Toyota Friend, Nike+, Xlively, Social Web of Things, Evrythng, etc. have aimed at integrating IoT with a social-networking framework [Atzori et al., 2014].

1.2 Smart Manufacturing

Although the benefits of connected social assets can be harnessed across several sectors such as health care, traffic routing, sports, etc. the manufacturing sector provides sufficient complexity, possesses immense potential to impact human lifestyle and environment, and employs maximum resources. It thus encompasses significant scope to benefit from this wave of innovation and transform into a smarter version. The broad aim of the Smart Manufacturing project we have worked on is to deploy Digital Twins and IoT to create a social network of assets to enable intelligent decision making and operations planning in manufacturing enterprises.

The National Institute of Standards and Technology (NIST) defines Smart Manufacturing as the systems that are "fully integrated, collaborative manufacturing systems that respond in real time to meet changing demands and conditions in the factory, in the supply network, and in customer needs."

The future of manufacturing is a blend of digital machines, industry-ready software and customer interfaces. It is now shifting from *personalised production* to *manufacturing by mass*, and the next wave of technological breakthroughs like mass-scale 3D printing of small components, super critical spares and safety equipment is likely to arrive soon. Smart factories of future are a marriage of versatile manufacturing paradigms and technologies such as cloud computing, the internet, real-time analytics, advanced robotics, Machine Learning, etc. to enable the production of large scale customised items, without an increase in the cost of production. Industries in future will plan operations automatically, and the only role a human manager would play would be to pick models and algorithms from a store of pre-developed tools that best suit a need.

1.3 Manufacturing sector in India and future scope

The scope of contribution of Smart Manufacturing is significant especially for the developing economies like India, where the manufacturing sector forms a huge share of

Gross Domestic Product (GDP) and plays a major role in employment generation for both skilled and semi-skilled labour. In the case of India, the manufacturing sector is the second largest employer, accounts for about 16% of the GDP, and can thus transform the lifestyles of its populace. In the near future, the Indian manufacturing sector would face the responsibility to absorb a significant demographic growth and disproportionate expansion of working age population. The Government of India has appreciated this and has taken several initiatives, such as the ambitious target to make the manufacturing sector account for 25% of India's GDP by the year 2025. The Prime Minister of India, Mr. Narendra Modi, has pitched India as a manufacturing destination at the World International Fair in Germany's Hannover in 2015. Campaigns such as Make in India, for attracting foreign investors in the country, further strengthen the Indian Government's vision to transform India into a global hi-tech manufacturing hub. With these continuous efforts to boost the Indian manufacturing sector, India is set to become the world's fifth largest manufacturer by 2020. Global giants such as GE, Siemens, HTC, Toshiba, and Boeing are attracted by India's market of more than a billion consumers and increasing purchasing power, and have either set up or are in the process of setting up manufacturing plants in India. The implementation of the Goods and Services Tax (GST) has made India a common market with a GDP of US$ 2 trillion along with a population of 1.2 billion people, which is a big draw for the investors. Apart from these, Government of India has also launched Digital India campaign to develop an advanced digital countrywide infrastructure[INDIA BRAND EQUITY FOUNDATION, 2015, Fut, 2015, Now, 2017].

Clearly, there is immense scope, need and support for promoting the manufacturing sector in India. Manufacturing, in short, is key to India's future. However, there haven't been many efforts by the Indian industries for development and innovation towards smart manufacturing. Little efforts by a few companies are in their infant stage and have not produced any significant results.

Another challenge faced by the Indian manufacturing sector is the amateur state of its sensor technologies and embedded computing power. The ages old machines used in most of the manufacturing shop floors of India show no signs of intelligence and are unable to produce the required extensive data for realisation of Smart Manufacturing.

While India struggles with modernising its manufacturing facilities, many government and non-government organisations around the world, especially Europe, are taking steps to promote the rise of smart manufacturing practices. 'Horizon 2020' is the European Commission's initiative under which multiple smart manufacturing projects are being funded. UK government has been funding companies who're working on development of smart manufacturing platforms. The German government has also marked smart manufacturing in its 'High Tech Strategy' projects by the name Industrie 4.0.

Staying in tune with the latest developments, and making sure that India gets to the forefront of manufacturing technologies both in the region and worldwide, is what this project promises.

1.4 Contribution of this project towards the next wave of innovation

A sincere intention of making a valuable contribution in the global technological consortium has encouraged us and driven us to design a novel MAS architecture and develop optimised algorithms for the three industrial problems targeted. Our efforts have primarily gone to solve the major problems being faced in the industrial systems at present, using latest technologies and deriving from concepts related to Industrie 4.0.

With our MAS architecture proposing newer implementations of collaborative learning, we are confident that it can serve as a radical innovation for Operations Planning, and Prognostics and Health Management (PHM) in manufacturing scenarios. Having developed a new method for calculation of IF and devised a novel distributed algorithm for operations planning, integration of the same with large-scale interactive systems can provide drastic reduction in processing times for job scheduling. Adaptation of a latest Long Short-Term Memory (LSTM) based prognostics algorithm to real-time, censored as well as uncensored, data, promises vital changes in the approaches in use today. Adding to these, the research done in this project for PHM of 3D Printers and development of a fundamental diagnostics model for the same, has allowed us to explore new pathways in this previously unchartered territory.

Social Internet of Things and Smart Manufacturing being the primary drivers of the next technological revolution, employment of various techniques from these concepts give our work a highly optimistic scope in the near future.

Chapter 2

A Multi-Agent Systems architecture for Social Industrial assets

2.1 Literature Review

Many paradigms have emerged to satisfy the requirements, and the challenges, of "new manufacturing" practices. Among which, agent-based manufacturing systems (Multi Agent Systems (MAS)), and Holonic Manufacturing Systems (HMSs) have received a lot of attention in academia and industry. MASs enable social behaviour of intelligent entities, through the capabilities of the agents forming the System. They are a broad software approach, unlike the manufacturing-specific approach of HMSs, focused on distributed control [Giret and Botti, 2004].

Decentralised architectures allow complex tasks to be divided into sub-tasks, allotted to the best suited agents. Decentralisation presents advantages like system robustness and agility, and elimination of data transfer lags. Enabled by MAS, these architectures have been used to tackle industrial problems. For example, [Giordani et al., 2013] make use of a MAS approach to tackle the problem of production planning and scheduling. A two-layer hierarchical approach is employed by [Mönch and Drießel, 2005] to decompose the scheduling problem into simpler sub-problems. [Christensen, 2003] proposed an architecture where agents focus on deliberative tasks on a higher level, while lower-level agents focus on real-time constrained control tasks.

[Bagheri et al., 2015] presented a step-wise approach to design a CPS architecture for an Industry 4.0 environment, and an adaptive clustering method for self-aware machines. [Bagheri et al., 2015] discussed how one should progress from the smart-connection level to the configuration level while designing an architecture. Drawing from Bagheri's ideas, in this paper we present an architecture which can be implemented for any industrial system inspired by the Social Internet of Things paradigm. We also introduce the concept of 'Virtual Assets', an additional agent layer aimed to standardise the data flowing into the asset's Digital Twins. This standardisation of data makes it possible for us to have a generic model of Digital Twins, thus eradicating the need to tailor different Twins for every other asset in the industrial system.

2.2 Learning in Multi-Agent Systems

Agents in an MAS need to keep learning in order to adapt to a dynamic environment. It has been shown that multi-agent learning can be reduced to single agent learning by considering the other agents in the system as part of the agent's environment. However, this may not always lead to an optimal solution and coordinated machine multi-agent learning becomes more important [Alonso et al., 2001]. An efficient way to achieve collaboration is to integrate social networking concepts into the Internet of Things. The assets in such a "Social Internet of Things" behave like social entities, sharing data and collaborating with one another to generate an optimal enterprise level solution. This paradigm, achieved by developing trust among assets which are "friends" with one another, permits navigability even after the point when number of nodes increase above those encountered in the traditional internet [Atzori et al., 2014]. Collaboration among assets in a system not only increases the responsiveness of the system, but also allows unseen events of importance to be broadcasted within a group of friends. This improves the accuracy of underlying algorithms by making a richer data-set available for training and prediction purposes. [Ning and Wang, 2011] describe an analogy of such systems with the social organisation of humans as: "each 'unit' human has its nervous systems made up of the same physical components and operating laws, but individuals possess their own sophisticated and unique consciousness and behaviour".

Multi-agent collaborative learning can be implemented through several kinds of algorithms: social algorithms, swarm intelligence, etc. An example of social algorithms is evolutionary computation, a kind of Stochastic Search method among reward-based learning techniques [Alonso et al., 2001]. Other approaches that incorporate collaboration between agents are swarm intelligence techniques, that try to emulate the efficiency of foraging seen in natural systems such as those of bees or ant colonies. [Panait and Luke, 2005]'s survey provides an interesting example about cooperative foraging, where agents are robots whose objective is to discover rock samples in an area and bring them to a specified location.

A relevant part of current research focuses on harnessing the technology capabilities by merging collaborative learning and Multi Agent Systems for the IIoT. However promising this idea may seem, there is a lack of an industrial-system architecture capable of integrating the social networking concepts with the IIoT. Our architecture addresses this gap by providing clearly organised levels, each suited for different analytics algorithms. Ours is a MAS architecture based on the SIoT paradigm, capable of being implemented on various industrial systems.

2.3 The Architecture

The objective of the MAS architecture developed in this project, which consists of three layers, is to implement Collaborative Learning for social industrial assets. The first layer is formed by Virtual Assets, software components that ensure that the data originating from machines is pushed to digital agents in a standardised format, and at regular intervals. The second layer consists of Digital Twins, digital agents that run the algorithms of interest for the asset manager using the standardised data from the Virtual Assets. The third layer of our architecture is the Social Platform, that can be hosted in a central server or in the cloud. All the communications to and from an agent, and the interactions with the external world, happen via the Social Platform (see figure 2.1).

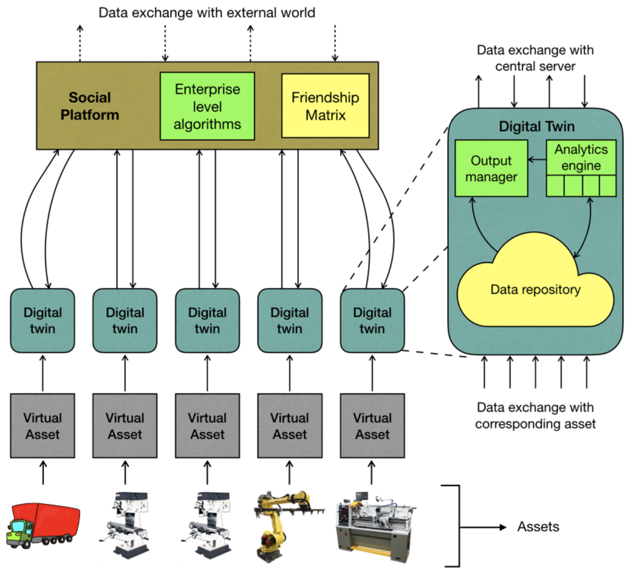

Figure 2.1: Schematic layout of the MAS architecture.

2.3.1 First Layer: assets, Virtual Assets and standardisation of data

In our architecture, the data originating from a physical asset is standardised by a Virtual Asset before being sent to the Digital Twin.

The motivation for the introduction of Virtual Assets is the heterogeneous nature of industrial asset fleets. A manufacturing facility, for instance, may have a milling machine, a packaging machine, and a lathe among many other kinds of machines. They might also come from different manufacturers, serve different purposes, and have different specifications. In another example, an automobile company produces different models of vehicles, which are suitable for different terrains or performances required. The number and types of sensors, or operating conditions, may vary among the vehicles. Virtual Assets are designed to standardise the data coming from these vehicles in a format that is conducive to a generic Digital Twin.

Virtual Assets

A Virtual Asset is a software component present for each corresponding physical machine. It is responsible for standardising the asset data before it reaches the Digital Twin. The data from Virtual Assets consists of three main parameters: the Machine Identifier, the Features (with time at which they were recorded), and the Events (kind of event, and time of event). The 'Machine Identifier' gives the asset a specific identity in the asset fleet, and includes information of the asset make, location and operator. 'Features' here refer to sensor generated values. 'Events' can be failures, warnings, user messages, etc. figure 2.2 describes how the data is made into a standardised format after passing through a Virtual Asset.

Similar enterprise-level solutions exist. For example, MTConnect [Vijayaraghavan et al., 2008], which standardises the data being transferred across the entire system. Our approach differs from MTConnect, as here the data from each asset is standardised individually by the Virtual Assets and not at fleet-level.

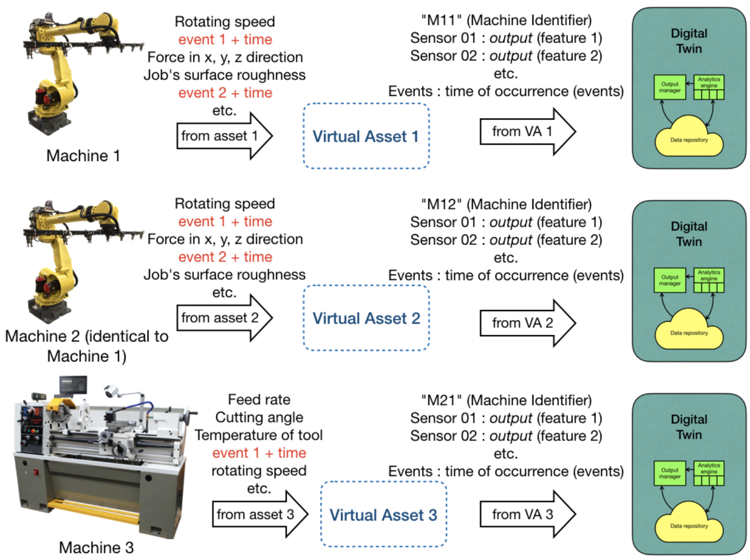

Figure 2.2: Virtual Assets and Data Transfer to Digital Twin

2.3.2 Second Layer: generic Digital Twins

Extensive instrumentation, increased digitisation and the heterogeneity of manufacturing systems make the design of industrial agents a difficult task. Even in relatively homogeneous asset fleets the make of the assets varies, and so does the nature of the asset's data. Addressing this challenge, the concept of a Virtual Asset presented in section 2.3.1 enables the development of a generic Digital Twin. These generic Twins are capable of working with standardised data provided from a variety of assets. This is aimed to free the asset manager from the cumbersome task of designing a specific Digital Twin for each of the many kinds of asset present in a typical industrial environment.

A Generic Digital Twin

The layout of the Digital Twin proposed here is generic, which means, it can be adapted to practically every type of asset and industrial problem. Figure 2.1 shows the layout of the proposed Digital Twin. Data flows into the Digital Twin from two sources: its corresponding asset, and the Social Platform; and is stored in a data repository. This data is then used to run a diverse set of analytic algorithms. These form the Analytic Engine of the Digital Twin. Typically for each *kind* of asset, their associated Digital Twins may run different kind of algorithms. Which algorithms will the Twins run is determined by the needs of the asset manager and conveyed by the Social Platform. An output manager monitors the streams of data flowing out of the twin, and is thus responsible for data sharing and collaboration with its friends in the asset fleet.

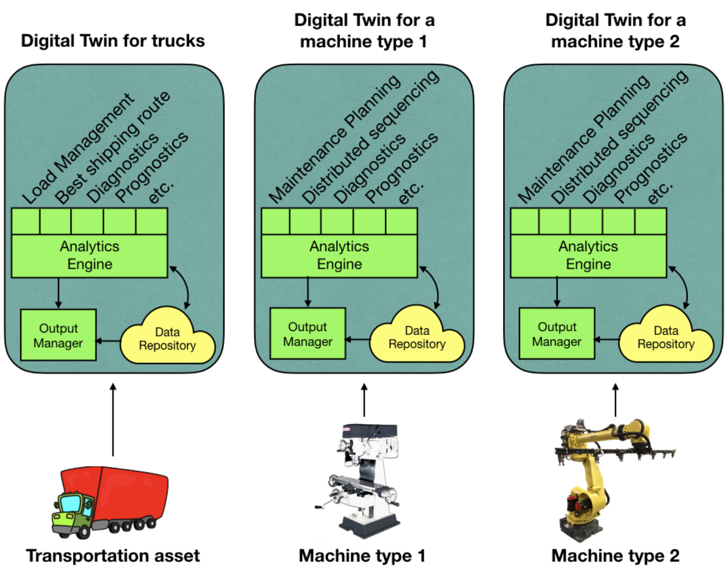

Figure 2.3: Role and working of the Digital Twin

The algorithms run by the analytic engine of the Digital Twin may address tasks like health management, performance optimisation, and other optional features which may be particular to that asset type. The figure 2.3 shows how prognostics and diagnostics are performed for all the assets, but certain tasks like load management and path determination are performed for the transportation assets only. Additionally, algorithms supporting a hypothetical centralised clustering performed by the Social Platform can be implemented in the analytics engine here. For example, a secondary hand-shake distributed clustering algorithm can be implemented on agents to reinforce centralised clustering.

The computing capabilities of the agents allow for a flexible heterarchy of the system. For instance, when a system is in operating state, the algorithms in the Digital Twins keep processing data both from themselves and from collaborating assets. This allows the algorithms (often designed to infer empirically-based models) to learn in real-time. This automatised heterarchy can be stopped at any time by request of the Social Platform.

2.3.3 Third Layer: the Social Platform

The Social Platform forms the third layer of the proposed architecture. Hosted in a single or multiple servers in the cloud, the Social Platform is both a gateway for human-Digital Twin interaction, and also an enabler for asset to asset communications. The primary capabilities of the Platform are shown in figure 2.1. These are running enterprise level algorithms which are implemented using data provided by the whole asset network, and storing the relevant system data in a repository. Algorithms implemented on the Platform are focused on performing enterprise level optimisation, and extracting fleet performance trends.

Collaboration among assets becomes efficient when an asset prioritises the data originating from its friends (similar assets). To enable this in our architecture, a matrix comprising of distances (similarities) between assets is formed and stored in the Social Platform. We call it the 'friendship matrix'. As the system operates, inter-asset similarities are calculated at regular intervals, subsequently updating the friendship matrix. Similarity may be calculated based on a variety of indicators such as feature data, machine type, environmental data, etc. Since it is a common channel for the data flowing in the MAS, the Platform theoretically is best informed to calculate similarity metrics. This is done through Enterprise level algorithms such as k-means clustering. Otherwise, decentralised clustering can also be run in the analytics engine of the Digital Twin. The information regarding the asset's clusters will in any case be in stored in the Platform's data repository, in form of a friendship matrix.

For each asset, its cohort of collaborating assets is given by the N closest assets in the friendship matrix. Collaborative learning is then implemented by sharing data between pairs of "friends". The data received by an asset from a friend may be weighted in the algorithms running in the analytics engine of the Digital Twin according to their estimated similarity.

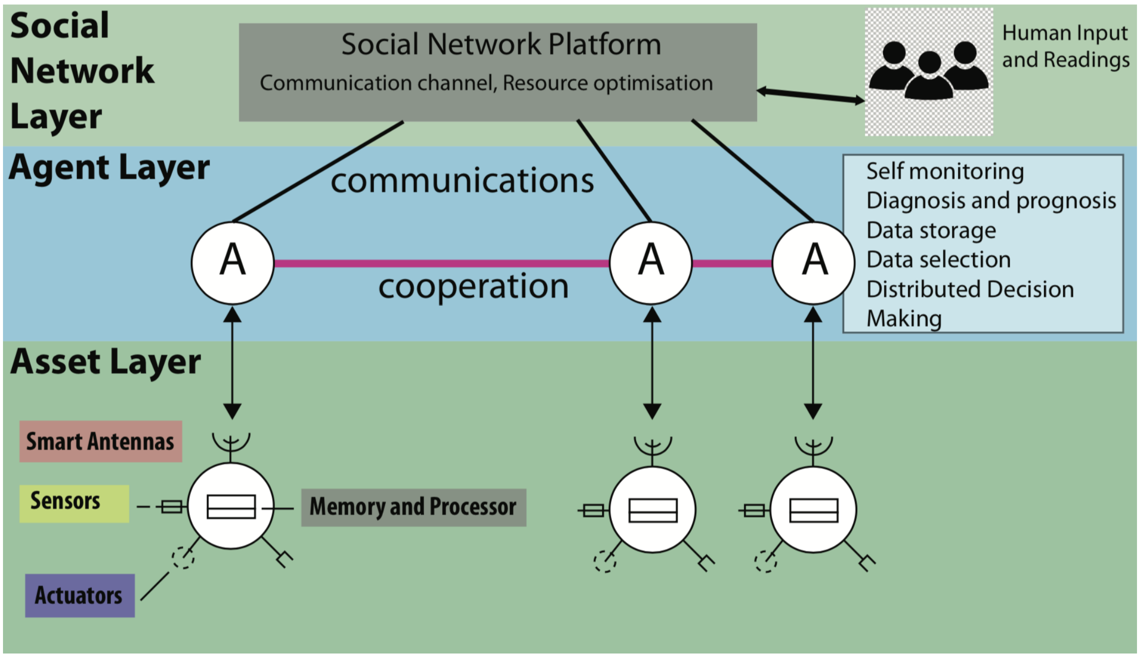

Figure 2.4: A Social Network of Assets

2.3.4 Collaborative Learning among Assets in a Multi-Agent System

In a fleet of assets a rare catastrophic failure may occur only to a small subset of assets. In this case, it would be beneficial to convey the information regarding this failure to the other similar assets in the fleet. If this is not done, we might face a scenario where an event, although already known to the fleet, would be unknown to machines which have not encountered it yet. Thus their algorithms will fail to predict it. This example becomes especially relevant for new machines being added to a fleet of old ones. In a collaborative MAS architecture, the trajectories corresponding to a newly registered event can be shared among similar assets and other agents can thus be made aware of such circumstances in future.

To make this inter-asset collaboration efficient, it is crucial to ensure that the data being shared covers all relevant information, and at the same time, is not bulky. To achieve this, the assets keep sharing certain pre-defined performance parameters at regular intervals while the asset is in normal condition. As soon as an asset encounters a certain *new event of interest*, the data corresponding to that time frame, i.e. a trajectory to that event, is shared as a 'new training data-set' for other assets. Subsequently, specific causes and analysis of the event are shared among Digital Twins too.

Apart from making the system more robust, collaborative learning makes a system agile and more efficient. For instance, through collaboration, machines of a manufacturing unit can actively manage their load, by continuously evaluating their health condition and comparing with that of other similar machines. A healthy machine is capable of producing more output, thus reducing the production load of deteriorated machines and the maintenance downtime.

2.4 An Illustrative Example: clustering and prognostics in the C-MAPPS data-set

We demonstrate the use of the above described architecture to determine the Remaining Useful Life (RUL) for a fleet of turbofan engines. Due to its link to Condition Monitoring, calculating an asset's RUL is a problem that combines asset management and IIoT technologies. Here, we use the C-MAPSS ([Saxena and Goebel, 2008]) data-set to showcase collaborative RUL estimation. The data-set consists of four fleets of engines, which are labelled as FD001-4. For our example we'll be using fleets FD001 and FD003 only. Engines within FD001 and FD003 share the same operating conditions, with the difference being that engines in FD003 fail due to High Pressure Compressor degradation and fan degradation while engines in FD001 only present the first kind of failure. The data-set employed here consist of multi variate time-series in the form of rows of sensor data recorded after fixed time-steps. Each machine starts normally, develops fault during operation, this fault grows in magnitude and the machine eventually fails. Both FD001 and FD003 feature 100 independent trajectories to failure. We group 20 trajectories to failure together, in each fleet, to simulate multiple machines. This is not ideal, but since all the machines in a fleet are identical, it is sufficient to serve as an example. Collaboration is implemented simply by sharing failure trajectories among clusters of machine 'friends'.

Sensor1	0.9	0.7	0.55	0.99	1	0.3	0.9	0.71
Sensor2	0.87	0.77	0.99	1	1	0.2	0.88	0.77
Sensor3	1	0.2	1	0	1	0.1	1	0
RUL	5	4	3	2	1	0	?	?
Class	2	2	1	1	0	0	2	2

Table 2.1: Example of a fixed-window classification for Remaining Useful Lfe. In green, classes assigned *a posteriori* of the known failure (marked in red). In blue, classes predicted by the classification algorithm.

For our case, Virtual Machines directly read the data of their corresponding engines from csv sheets, instead of receiving data from a real asset. To simulate real-time operation, the VMs read the data at fix time intervals, and are unaware of what lies ahead. The data, after being pushed to the Digital Twins using the 'socket' library, is processed by a naive prognostics algorithm, based on a fixed window K-Neighbours classifier from the 'sklearn' machine learning library. In short, the data coming from the asset's sensors is classified according to its known remaining useful life based on the width of a predefined time window (see Table 3.4). This classification algorithm is then used to make approximate predictions about the RUL of new trajectories.

Chapter 3

Operations Planning in a Manufacturing Shop Floor

3.1 Introduction

With the rise of Internet of Things (IoT) and surge in the number of disruptive innovations in embedded technologies in the past decade, it only feels apt to revise the conventional approaches being applied in Manufacturing and Industrial scenarios to enable them to employ these latest technologies and benefit from their advantages. Today, we have at our disposal tremendous amount of computational power, greatly improved sensing technologies along with fast and reliable storage facilities. Combined application of these to real-world problems in recent years is what has led to the popular technological paradigm known as Big Data. The growing need for data-dependent solutions and self-prudent assets in Manufacturing shop floors, to counter the inefficiencies in current approaches, and the possible utilisation of Big Data and IoT technologies for the same, is what serves as our motivation to test our MAS architecture in a Manufacturing shop floor environment.

After a comprehensive review of existing literature, we identify the Operations Planning and Job Scheduling problem as the most suitable to obtain evidential results to bolster our architecture. For parallel machines manufacturing shop floor scenarios, the planning algorithms implemented today face limitations in terms of the vast deluge of data and processing times. The reason for such issues can be understood after realising that these algorithms employ a centralised approach and are based on hierarchical architectures. Our emphasis on distributed computing and decentralised systems arises after finding the significant improvements for the same, described in the following subsection.

3.1.1 Centralised v/s Decentralised Systems

Where conventional systems reply on a single computing server, which processes data being pushed from assets across the network, the distributed systems harness the benefits of modern computing capabilities. With the powerful and miniature microprocessors available today, each asset in an industrial system can be provided with its own computing capability, responsible for analysing the data generated from the corresponding asset. Following are the benefits of decentralised systems over the centralised systems:

Agility

In a distributed system, the local agents analyse the data arising from the corresponding assets. This makes the system way more agile than the conventional centralised approach where the asset data is sent to the central server which is solely responsible for processing data from every asset.

Data Transfer Lag

Another advantage of processing the data at its source is the elimination of time lag encountered while transferring large amount of data from the assets to the central server. Owing to this ability of quickly responding to environmental changes, an asset can independently respond in an emergency situation, such as that of a sudden failure of some critical component, rather than waiting for a decision from the central server.

Flexible distribution of decision making power

Since the distributed systems have computational power at every level- machines, departments and the central levels, the decision making power can be assigned to the agents best suited for it.

Robustness

Lastly, distributed systems are significantly more robust than the centralised systems. In a distributed system, the failure of any single asset does not affect the functioning of other assets, unlike the centralised systems where the failure in the central server can disrupt the entire system.

We feel our architecture is best suitable here, for being implemented on a manufacturing shop floor, and would be an upgrade from the existing centralised approaches. Figure 3.1 is a diagrammatic representation of how we basically implemented it. Section 3.2 describes the problem statement and the sections following it are devoted to a numerical example and results obtained for this particular implementation of the architecture.

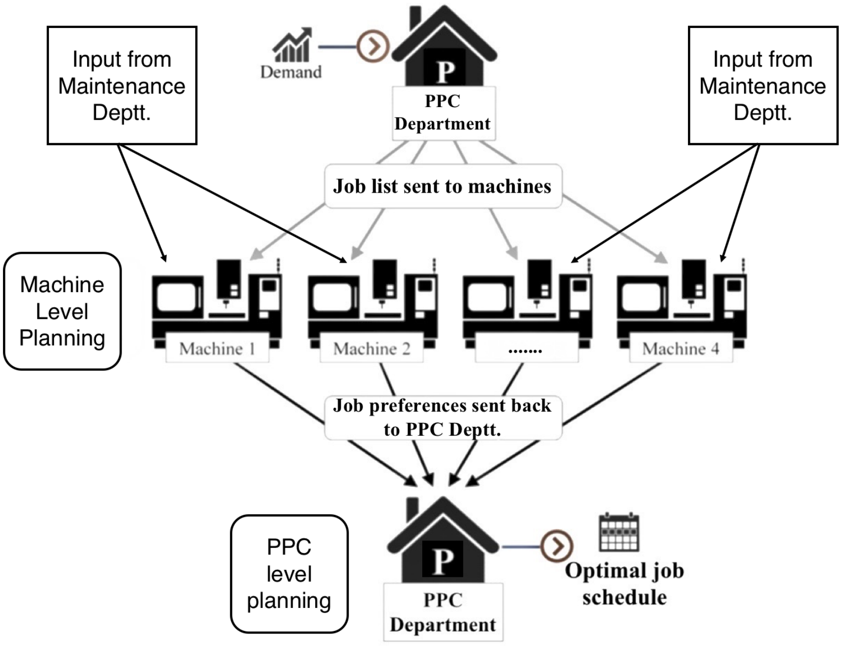

Figure 3.1: Distributed Sequencing

3.2 Problem Statement

In our implementation, we have simulated a parallel machine system for which sequencing of jobs is to be optimised. We'll be relying on distributed computations to achieve this. The problem is described as follows:

Consider a shop floor with M parallel machines (single component machines) and a demand of N jobs. Initial age of machines is assumed to be different and the time to failure is calculated by assuming a 2-parameter Weibull probability distribution function. Machine failure is of only one type: immediate breakdown. Each job requires its process to be done by a single operation on a single machine only, and all machines are assumed to be capable of performing all kinds of operations required. Hence a job, while sequencing, can be sent to any machine without compromising its processing time, or quality.

Since we are focusing on sequencing, we do not consider any unplanned failures. However, we do consider Preventive Maintenance while formulating the sequence. Preventive Maintenance (PM) is seen as a job (PMJob) having a constant processing time which is added when the reliability of the machine falls below a specified threshold. After a job is sequenced on a machine, we calculate the machine's reliability (from Weibull distribution) and compare it with the threshold set for PM. Threshold we consider here for adding a PMJob is at a reliability value of 0.5.

Preventive maintenance (PMJob) has associated with it, a restoration factor (α) and maintenance downtime for that machine. Restoration factor implies restoration of α percent of that machine's age. We describe the distributed sequencing approach with the help of a pseudo-code (Figure 3.2). A flow-chart (Figure 3.3) is also attached to explain the algorithm in a lucid manner. The complete environment including the PPC department and the machines, is developed on Java platform.

Procedure: Scheduling at local machine:

begin

- get all possible job sequences using Directed Acyclic Graph (DAG) dynamic job scheduling algorithm
- sequences are sorted according to their IFs in decreasing order
- send the list of schedules, with their IFs and machine IPs as a sequence packet

end

preparing an enterprise level schedule:

Received sequences from all machines sorted according to decreasing IFs

begin

- make a list containing all sequences and sort it in decreasing IF
- assign first sequence to the corresponding machine
- eliminate all the sequences containing the machines and jobs already assigned
- repeat till all the jobs are assigned

end

Steps followed by machine agent:

1. Get all the jobs from the scheduling department, sorted in their increasing order of processing times.
2. The agent then searches for, and calculates IF for all the possible job sequences, starting from only one job till maximum possible jobs in the machine shift duration.
3. An IF (calculation explained below) is calculated for each sequence.
4. A final list of job sequences, each with a corresponding IF, and sorted according to descending IFs is sent to scheduling department.

Steps followed by PPC agent:

1. The scheduling department first receives job sequences from all machine agents, and sorts them in descending order of IF.
2. Sequence with maximum IF is first incorporated into enterprise schedule, and the sequences with repentance of jobs and machines are removed.
3. A final job sequence, distributed to all machines is thus made and is sent to maintenance department.

Figure 3.2: Pseudo code for distributed sequencing of jobs

3.3 Calculation of IF

When a job demand reaches the machines, the machines generate all possible job sequences for their shift duration and send them back to the PPC. But how does the PPC know which sequence is best suitable for a particular machine?

To address this, we have developed an Intensity Factor (IF), which is an indicator to the priority which must be given to a particular sequence. IF is calculated based on the lateness of production for the sequence. When a machine sends a list of all possible job sequences to the PPC, it also calculates an IF corresponding to each of the sequence and sends it together with the list of job sequences.

IF is calculated according to the following equation:

$$IF = \frac{L_{max} - L_{seq}}{L_{max}} \tag{3.1}$$

where L_{max} is the maximum lateness encountered, i.e. when no job is processed for the entire shift duration, and L_{seq} is the lateness encountered for the particular sequence considered. Calculation of L_{max} and L_{seq} is as shown below, where t_d is the due tine for the job, t_c is the time of completion of that job, and t_s is the machine shift duration. $N =$ total number of jobs

$$L_{max} = \sum_{n=1}^{N} (t_s - t_d) \tag{3.2}$$

$$L_{seq} = \sum_{n=1}^{N} (t_c - t_d); \tag{3.3}$$

$t_c = t_s$ *for jobs which are not processed in the considered sequence*

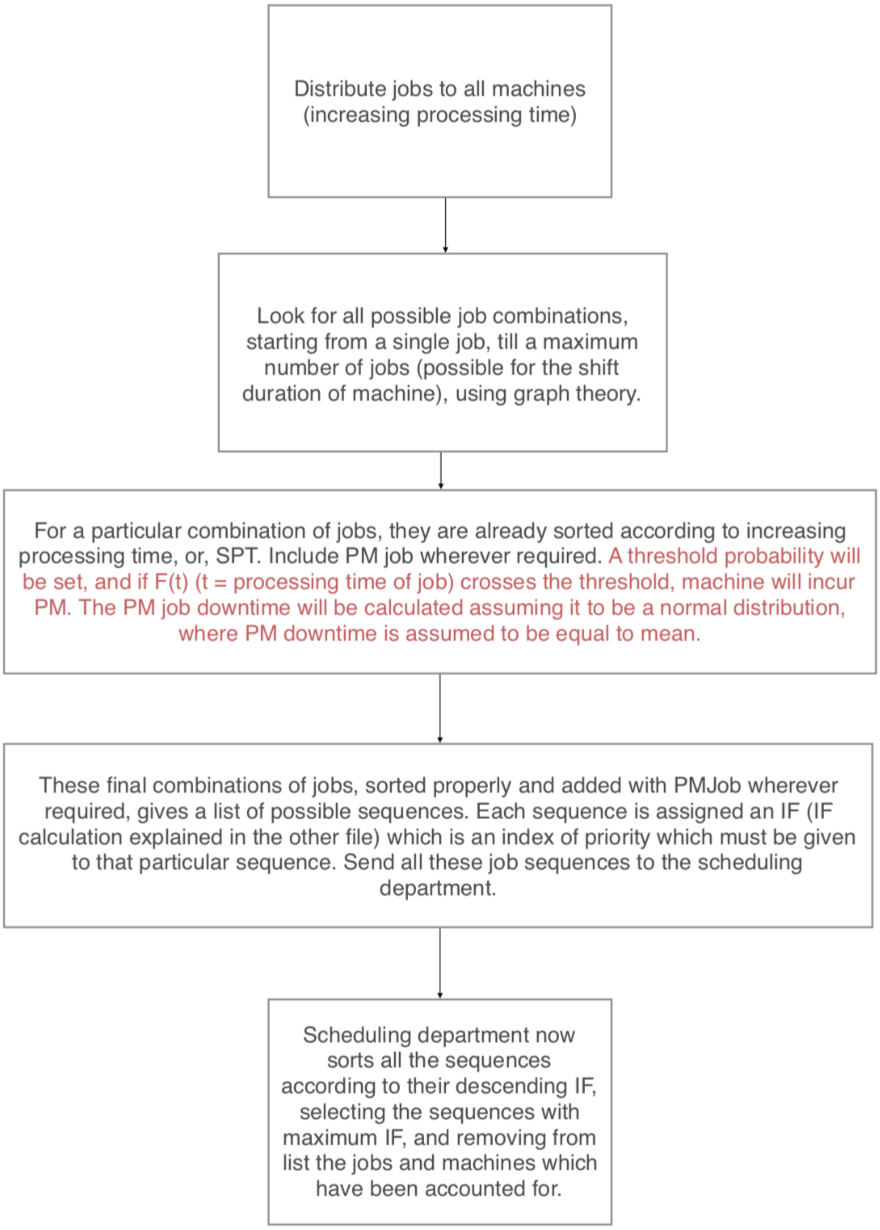

Figure 3.3: A flow-chart describing the proposed approach for distributed sequencing in a Manufacturing shop-floor

3.4 Numerical Example

To illustrate the proposed approach, a numerical example is shown, which involves sequencing 9 jobs into 5 machines. The machines we use here are all different, in terms of health/ initial machine age. However, all are identical, when it comes to the ability to process jobs. We take the Weibull parameters for time to failure of machines as $\beta = \mathbf{2}$ and $\eta = \mathbf{1000h}$. time to carry out PM is **8h** and the restoration factor is **0.6**.

Jobs to be scheduled are also of varying processing times, but as mentioned above, any job can be sequenced to any machine, without a compromise in job quality or processing time. Processing times for the jobs are constant. All parameters defining machines used and the jobs to be sequenced are shown in tabular form below:

PMJob Time	8 h
Restoration Factor	0.6

Table 3.1: PMJob Details

Jobs	**Processing Time (in *Hours*)**	**Due Time (in *Hours*)**
J1	55	260
J2	85	260
J3	205	260
J4	105	260
J5	155	260
J6	185	260
J7	225	260
J8	135	260
J9	225	260

Table 3.2: Jobs Description

Machine No.	**Age**	**eta (η) (in *Hours*)**	**beta (β)**
M1	0	1000	2
M2	973	1000	2
M3	1969	1000	2
M4	2319	1000	2
M5	2497	1000	2

Table 3.3: Machine Data/ Parameters

Our approach here relies on distributed computing for generating an optimised solution. Job list, or the customer requirement is first received at the PPC. PPC initiates the job scheduling process by first sending the job list to machines in an increasing order of their processing times. This is done in order to eliminate the unnecessary iterations later while generating all possible sequences at machine stage. Subsequent steps are clearly explained in the flow-chart (figure 3.3)

3.5 Results

The graph in figure 3.4 shows the final results of our experiment. Different cases on the horizontal axis are the different combinations of jobs and machines we had considered. Case 1 is the most basic 2 Machines-2 Jobs, and the Case 9 is 5 Machines-9 Jobs. Cases 2 to 8 are with intermediate complexity. Figure 3.5 shows the optimal job schedule for the Case 9 (5 Machines-9 Jobs) with the corresponding IF values for each sequence

Upon quantitative comparison of the processing times taken by the centralised and decentralised approaches for assigning job schedules to machines, we observe only a marginal increase in the processing times for our approach, countering the exponential increase in conventional centralised approaches, as the problem complexity increases (figure 3.4). Alongwith this, the scheduled job sequences are the most optimal for the data used, with newer machines taking up more jobs. This proves the efficiency and correctness of the algorithm developed by us.

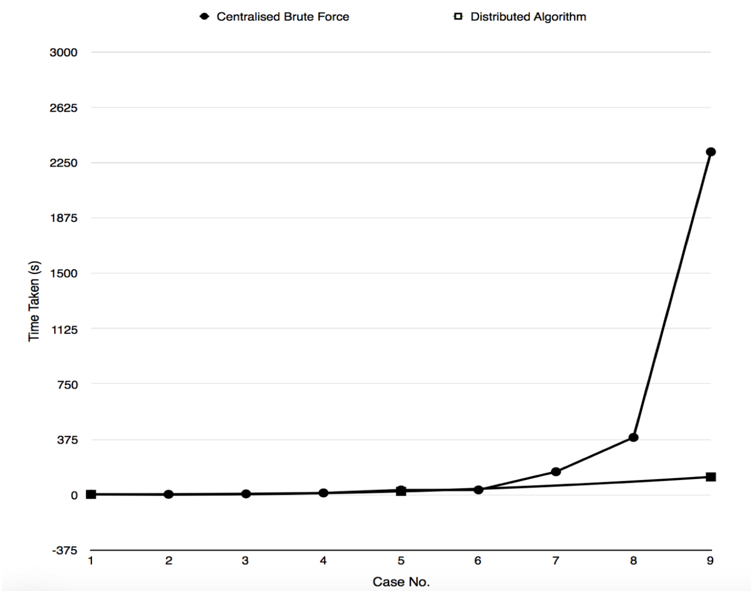

Figure 3.4: Comparison of conventional centralised and decentralised approach

Machines	Job Schedules and corresponding job processing time in hours	Intensity Factor (IFs)
Machine 1	J1: 55 ; J2: 85 ; J4: 105 ; J8: 135	2.483
Machine 2	J5: 155 ; J6: 185	1.296
Machine 3	J3: 205	1.168
Machine 4	J9: 225	1.149
Machine 5	J7: 255	1.122

Figure 3.5: Result for the case of 9 Jobs and 5 Machines

Chapter 4

Prognosis of NASA Turbofans (C-MAPSS data-set)

When we talk about Smart Manufacturing and a collaborative learning environment, we certainly imply providing the assets with intelligence. When referring to different physical assets such as gas turbines, jet engines etc., the concept of "smart" assets is necessarily intertwined with the prevention of asset failure and prediction of events for the near future. Over their lifespan, assets suffer deterioration both in their intrinsic operational capabilities and their efficiency compared with newly designed assets. Thus, in order to optimise maintenance policies, one requires some knowledge of the time to failure of the particular assets under scrutiny. This can be obtained by individual or fleet statistics and by measurement of the current state of the asset either though inspection or through continuous monitoring by digital sensors. In our work, we adapt a latest Deep Neural Networks based prognostics algorithm known as WTTE-RNN (Weibull Time to Event - Recurrent Neural Network) and implement it in a distributed manner. A fundamental test is then done for collaborative learning in this approach.

This part of the project was done at the Distributed Information and Automation Laboratory (DIAL), Cambridge under the guidance of Ajith Kumar Parlikad.

4.1 Problem Statement, Data-set and Network Setup

We define the problem by considering a fleet of related assets, jet engines here, and implementing the WTTE-RNN algorithm [Martinsson, 2016] on the censored as well as uncensored data obtained from these assets in a distributed manner. Once this is done successfully, similar assets from the fleet should collaborate with each other and this should result in better accuracy of the predicted times. The description of the problem here is very simple: given the readings of hundreds of sensors, predict when a fault event would occur.

To simulate the same, we have used the standard NASA C-MAPSS jet engine failure data set, and built different types of assets from chunks of trajectories from the 4 distinct types of data (FD001, FD002, FD003, FD004), as shown in the figure 4.1.

For building a network of such machines, we have implemented socket programming and used the resources at DIAL, Cambridge, to develop the Virtual Assets and the Digital Twins for several machines. The complete architecture is developed on Python 3.5 platform.

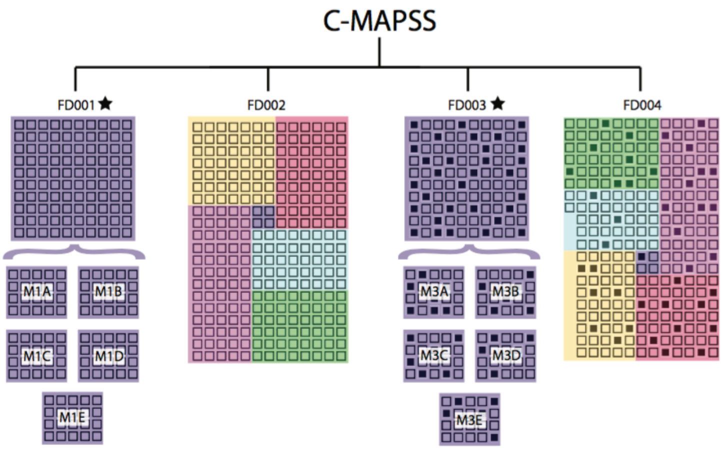

Figure 4.1: C-Mapss data-set used and simulation of machine fleet

4.2 The Algorithm

Egil Martinsson, a Swedish researcher from the Chalmers' University [Martinsson, 2016] recently proposed a novel prognostics algorithm that combines deep neural networks and Weibull distributions, known as the Weibull Time to Event - Recurrent Neural Network (WTTE-RNN) algorithm. After realising the novelty of this approach and use of latest algorithmic techniques, we decided on this as the best for event prediction for the NASA data-set. This uses a recurrent neural network in order to maximise the following log-likelihood function:

$$log\Big(\mathcal{L}\Big) = \sum_{n=1}^{N}\sum_{t=0}^{T_n}\Big[u_t^n log[Pr(Y_t^n = y_t^n|x_{0:t}^n)] + (1-u_t^n)log[Pr(Y_t^n > y_t^n|x_{0:t}^n)]\Big] \tag{4.1}$$

This choice for the log likelihood may appear obscure but in fact it is easy to understand. Let's first take a look on the two probabilities that are to each other for each summation term. The first probability to appear is $u_t^n log[Pr(Y_t^n = y_t^n|x_{0:t}^n)]$ which simply means: *in case of uncensored data, where we are aware of the real time to failure* ($u_t^n = 1$), *maximise the probability of our predicted time to failure* Y_n^t *being equal to the real time to* *failure y_n^t given the known values of the time series before time $t, x_{0:t}^n$. The second term, $(1 - u_t^n)log[Pr(Y_t^n > y_t^n | x_{0:t}^n)]$ means: if we are dealing with censored data, and have not yet observed the real time to failure ($u_t^n = 0$), maximise instead the probability of the predicted time to failure Y_n^t being bigger than the current time spent without failures y_n^t. $\sum_{n=1}^{N} \sum_{t=0}^{T_n}$ account for the summation over all the recorded failure trajectories and over all the time-steps of each trajectory.* [Salvador Palau, 2017]

A Long Short-Term Memory (LSTM) Neural Network is employed for optimising the above function. The LSTM model is trained using sample data to obtain a function that essentially converts multi sensor time series data to parameters of the time to event Weibull distribution:

$$f_{\vec{w}}\Big(x_{0:t}\Big) = \Big(a\big(t\big), b\big(t\big)\Big) \tag{4.2}$$

Where $x_0 : t$ is a matrix containing the sensor values in the first t time-steps and $\vec{w}$ indicates the weights of the LSTM Neural Network, which determine the function f. The α and β values obtained form a Weibull probability distribution function for the time to event.

The same function can be used to obtain the expected time to event, by simply using the mean of the Weibull distribution:

$$E[tte(t)] = \alpha(t)\Gamma(1 + \frac{1}{\beta(t)}) \tag{4.3}$$

The design of the LSTM Network used here is a 5 layer architecture as shown:

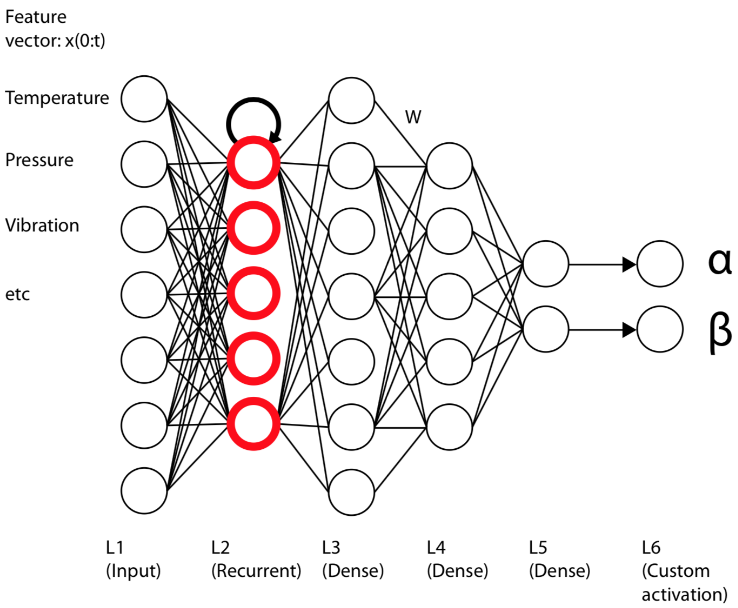

Figure 4.2: Internal schema of LSTM Network

4.3 Our Implementation

For the validation of our MAS architecture, we developed a network of assets similar to that in section 2.4. Figure 4.1 shows the division of the data-set into groups of trajectories to simulate engines. With the objective of imitating a real-time environment, we made the Virtual Assets send data to the corresponding twin at a rate of 10 time steps in an instant. At each instance of data, we first predict the time to failure corresponding to the complete data till that point, then re-train the neural network including this data. The correlation plots obtained here (figure 4.3) show the highly optimistic results obtained with a correlation of 85.2%. The real-time graph (figure 4.4) plots predicted times to failure at each instant and compares it to the actual times to failure. The failures here, as defined in the NASA Run to Failure data-set, can be of two types, with HPC degradation occurring in both FD001 and FD003 kinds, and fan degradation in only FD003 kind machines.

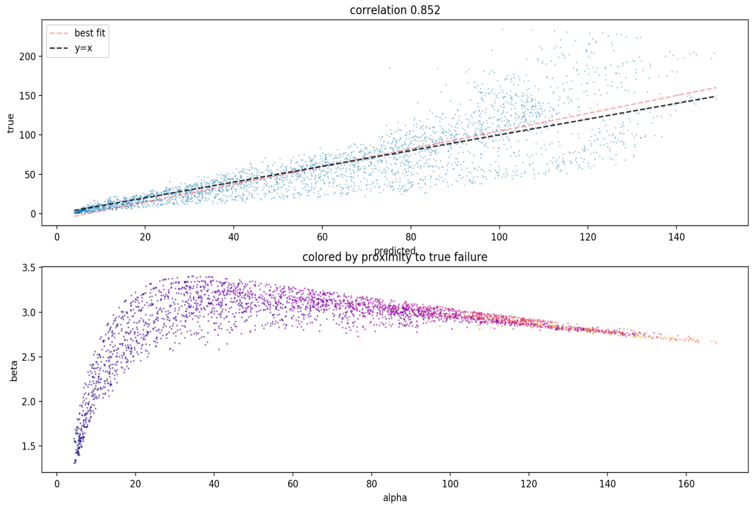

Figure 4.3: Correlation graph

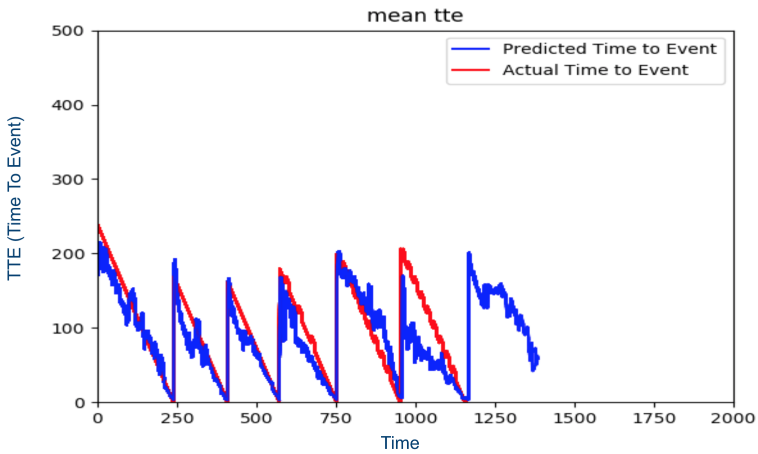

Figure 4.4: Real-time event prediction

Apart from the prognostics algorithm implemented in the Digital Twins, a basic K-Means clustering algorithm is implemented on the Social Platform, aimed to determine the friendship matrix. This algorithm, identifies and clusters similar engines based on Euclidean distances between the sensor values. This has been implemented to illustrate the role of the Social Platform and the Digital Twin's output manager. The output managers share the average of the data points received from their corresponding VMs in a time-step, with the Social Platform, which serves as a statistical indicator of the status of the asset. Then, the Social Platform uses these points to determine the Friendship Matrix of the assets using the centroid based clustering approach (figure 4.5). Once the clusters are stable, in each time step, the Social Platform then shares every asset's data with its friends, enabling every asset with large amount of data from its cluster.

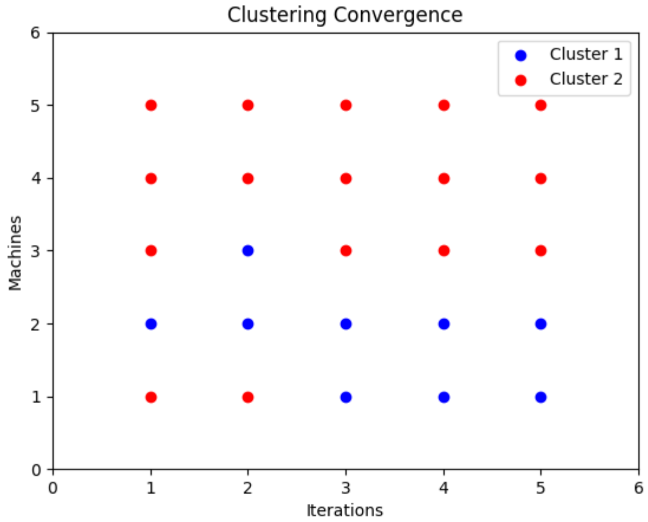

Figure 4.5: Clustering Convergence by the Social Platform

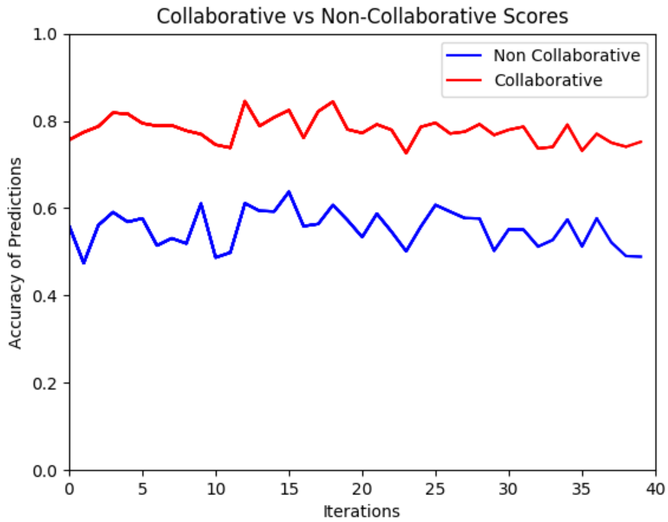

Figure 4.6: Accuracy of the fixed-window predictive algorithm for a machine of Cluster 1 in the Collaborative (red) and Non-collaborative (blue) cases.

After implementing the example described above and quantifying the results obtained for collaborative and non-collaborative approaches, we find the significantly higher accuracy of Collaborative approach as evidence for the efficiency of our architecture and proof of the advantages of collaborative learning (figure 4.6).

Chapter 5

Diagnosis of 3D Printers

5.1 Additive Manufacturing

With increasing global competitiveness and demand for mass customisation, the manufacturers today look out for different ways to produce custom goods at lowest possible costs. Having been refined over a hundred years, modern manufacturing practices have been revolutionised with technologies like automation, robotics, advanced computer-aided design, sensing and diagnostic technologies. The factories today are at a pinnacle of efficient mass production- producing items for mass consumption at lowest possible cost. However owing to numerous alterations involved, like setting up of new moulds or reconfiguring the equipment involved, the task of setting up the assembly line for producing a new item is cumbersome. This posts a serious impediment to producing small batches of intricate parts, or to item customisation and thus a major challenge faced by the manufacturing enterprises.

The additive techniques not only address the aforementioned challenge to mass customisation, but also put forth a plethora of added advantages- reduced production costs, greater control over the internal structure, geometry or the materials used, etc. and has thus seen rising trends in manufacturing practices all over the globe. In contrast to cutting down or moulding metals to the required shapes and sizes, the additive manufacturing techniques fuse together layers of materials using high temperatures and produce complex parts directly from a computer file.

The technology first came into existence when a group of engineers at MIT patented the "three-dimensional printing techniques" in the early 1990s, and has seen early uses for rapid prototyping and people producing tchotchkes as hobbies using their personal small-scale 3D Printers. But owing to the benefits it offers and the possibility of use in industries, 3D Printing has seen increasing use in modern manufacturing shop floors also. A classic example is the use of laser-based additive manufacturing to produce the engine fuel nozzles of one of the GE's best-selling LEAP jet engines. In contrast to the conventional techniques which involved welding 20 different parts together, this 3D printed nozzle is now a single unit. It offers a 25% reduction in weight, is 5 times more durable, and has 15% increased fuel efficiency.

This nozzle is a pioneer to the plethora of 3D Printed components in future. However, the real cause of excitement today is possibility of manufacturers to use this additive manufacturing practice in combination with the technologies like cloud computing, Internet of Things, the internet, real-time analytics, machine learning, etc. to transform conventional manufacturing shop floors into Smart Manufacturing and produce customised products on a large scale, cheaply and efficiently.

5.2 Health Management for 3D printers

In this part of the project, an attempt has been done to implement aforementioned architecture for collaborative prognosis of 3D printers in a laboratory environment, resembling a Smart Manufacturing shop floor. By deploying numerous sensors on the 3D Printer, we have simulated a smart machine which is able to see, feel and hear like humans. The data generated is in a way indicator to the events that might occur over next time-steps, and health of the printer. Further, we have successfully done a basic diagnosis experiment for comparing a healthy printer and a deteriorated printer. The diagnosis is done for 3 different components, the deterioration of which is gauged using the data acquired from the sensors we have installed.

The 3D printer used for experiments is the Prusa i3 single-filament model. The printers are all connected on a single server and are each assigned a Digital Twin, which processes the local data generated by implementing various analytic algorithms. The approach employed in this part of the project is further explained in the subsequent sections.

5.2.1 Sensors and Instrumentation

In the first step of our implementation, we conducted a Failure Mode and Effect Analysis (FMEA) for the printer at hand and identified the most critical components:

1. The extruder nozzle

2. The guiding rod and belt

3. The extruder spur gears

Figure 5.1 is the FMEA conducted and the proposed approaches.

Failure Modes and Effects Analysis for 3D printers			
Failure modes	**Causes of failure**	**Effects of failure**	**Detection methods**
Nozzle Blockage	Prolonged temperatures below the melting point of filament material Usually encountered when the filament material is changed, for example from PLA to ABS	Nothing gets printed, or a very uneven print	Increased power consumption of the extruder spur gears Detected by studying the temperature variations of the nozzle tip
Reduced friction on the extruder spur gears surface	Unmatched rpm of the gears and rate of extrusion of filament material from the nozzle tip. Initially, more filament is drawn than what is released. This causes the filament material to get accumulated on gear surface due to friction. The gear teeth are of no use and the rate of extrusion of filament material then slowly decreases. Blocked nozzle is an extreme case, where the gears keep spinning but since the filament is not able to flow, filament material sticks to the gear surface. Since the filament is not flowing, the gear surface heats up due to friction and the condition becomes worse with prolonged exposure	Accumulation of filament material between the extruder gears and the nozzle. Loss of control over the filament flow, and thus the job pattern	Comparing the flow of filament material with the rpm of gears. Studying the varations in the power consumption pattern of gears
Loosening of belts used for guiding the filament extruder	The dimensions of the belt does not match the one required by the guiding wheels Temperature variations of the belt due to over-use of the printer Belts not replaced after certain duration, which maked the surface uneven and may also decrease the thickness of the belts	Inability to control the extruder movement precisely Poor print quality	We measure the distance travelled by the extruder, using optics sensors, and compare it with the revolutions of the wheels driving the conveyor

Figure 5.1: Failure Modes and Effects Analysis for 3D Printers

After identifying the critical components and the relevant health indicators, the next step was to identify the suitable sensors. Selection of sensors includes a lot of considerations like their range, working environment, the sampling frequency, etc. After a lot of research, the following sensors were identified to be the best suited for our experiment:

1. **Temperature:** LM35 (For Nozzle temp.) and DHT (For Heat-bed temp.)

2. **Proximity:** Sharp 2Y0A21

3. **Vibration:** MEAS Piezoelectric sensor

Figure 5.2 shows our data acquisition setup, with exact positioning of each sensor marked.

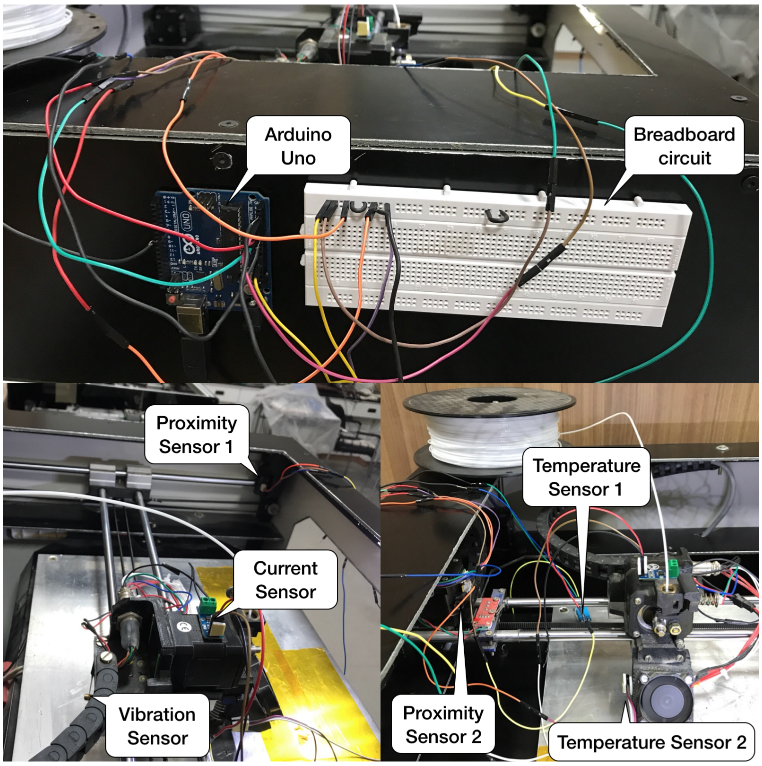

Figure 5.2: Data Acquisition Setup

5.2.2 Diagnostics model

Where the current diagnostics models are more focused and dependant on the quality and type of print, we have tried a model to gain information from the sensors during its run to identify the causes of the different faults occurring in the printer operation. After the deployment of sensors and data acquisition from the setup, the data obtained from the five sensors is used to design a diagnosis model for defining the state of the printer.

We take 20 data points in a single subset and take the Root-Mean Square (RMS) values of each subset to form the data for Condition Indicator (CI) of the printer [Yoon et al., 2014]. Once the CI table is computed, we use the 'sklearn' library of the Python 3.5 platform and develop a 10-fold Cross Validation Classification model, making use of the Support Vector Classifier (SVC) to classify the state of the printer, based on a data instance, as healthy or faulty.

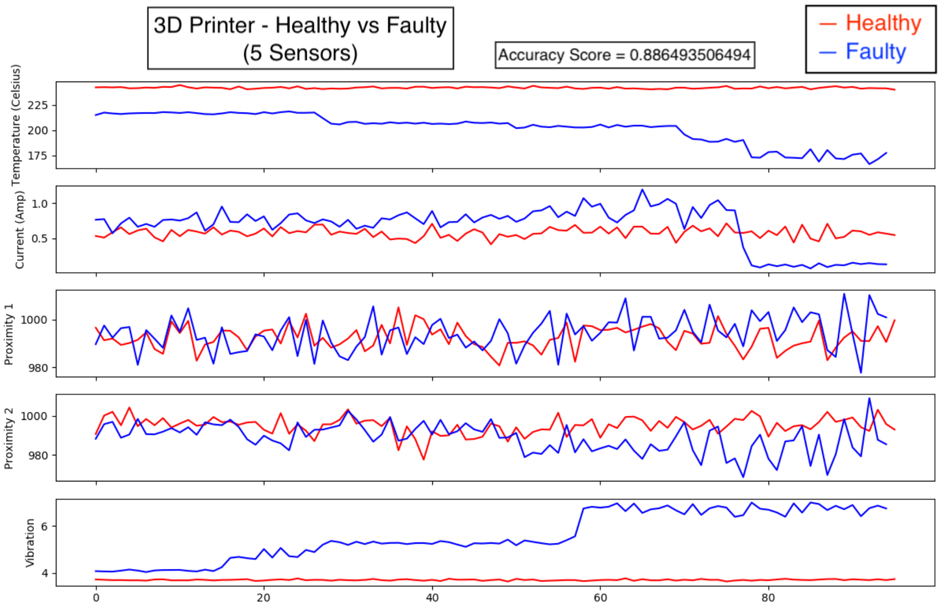

Figure 5.3: Diagnostic classification of healthy and faulty 3D Printers

After analysis of the data from the 5 sensors, we can infer that the temperature, vibration and current sensors are vital indicators of the trends regarding faults in the printers, in our setup. Upon training the classifier model with this data, predictions made on further data obtain a good accuracy score of 88.65%.

After some further validations, we hope to put forth a model that successfully determines the health state of the 3D Printer, and has a scope in prognosis of failures of its components.

Chapter 6

Conclusion

The problem identified and the corresponding architectures and algorithms designed, along with the extensive research done on the existing literature, as part of the B. Tech. Project can be successfully realised from the highly accurate and impressive results obtained upon validation of the architecture and the algorithms in the three different implementations.

6.1 Brief Summary

1. Novel MAS Architecture developed
2. Implementation on Operations Planning problem for a Manufacturing shop floor - New Distributed algorithm devised and efficient results obtained
3. Implementation on NASA Turbofans data-set for Prognostics - Adapted a latest prognostics algorithm and significant improvements observed on doing distributed and collaborative prognostics
4. Implementation on 3D Printers for PHM - set up the 3D printer for proper data analysis after deployment of sensors and developed a diagnostics model which obtained valuable results

6.2 Concluding Remarks

To conclude the entire work done as part of the project, our major emphasis on novelty and originality of work has aided us to develop a very promising MAS architecture with great scope in the future to alter the conventional design of industrial systems for the better. The implementations and all the promising results bolster the applicability of our architecture for the real-world industries.

A research paper on our project, in collaboration with the people at Institute for Manufacturing, University of Cambridge, has been submitted for the Special Session at the 16th IFAC Symposium on Information Control Problems in Manufacturing (INCOM) to be held in Bergamo, Italy in 2018.

Bibliography

[Fut, 2015] (2015). The next manufacturing destination - Make In India.

[Now, 2017] (2017). Manufacturing Sector in India: Market Size, FDI, Govt Initiatives — IBEF.

[Alonso et al., 2001] Alonso, E., D'inverno, M., Kudenko, D., Luck, M., and Noble, J. (2001). Learning in Multi-agent Systems. *The Knowledge Engineering Review*, 16(3):277–284.

[Atzori et al., 2014] Atzori, L., Iera, A., and Morabito, G. (2014). From "smart objects" to "social objects": The next evolutionary step of the internet of things. *IEEE Communications Magazine*, 52(1):97–105.

[Bagheri et al., 2015] Bagheri, B., Yang, S., Kao, H. A., and Lee, J. (2015). Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. In *IFAC-PapersOnLine*, pages 1622–1627.

[Christensen, 2003] Christensen, J. H. (2003). HMS/FB Architecture and its Implementation. In *Agent Based Manufacturing: Advances in the Holonic Approach*. Springer-Verlag Berlin Heidelberg.

[Evans and Annunziata, 2012] Evans, P. C. and Annunziata, M. (2012). Industrial Internet: Pushing the Boundaries of Minds and Machines. *General Electric*.

[Ganchev et al., 2016] Ganchev, I., Ji, Z., and O'Droma, M. (2016). Designing a low-cost data transfer unit for use in IoT applications. In *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 85–88.

[GE Digital, 2017] GE Digital (2017). The Digital Twin: Compressing Time to Value for Digital Industrial Companies. Technical report, General Electric.

[Giordani et al., 2013] Giordani, S., Lujak, M., and Martinelli, F. (2013). A distributed multi-agent production planning and scheduling framework for mobile robots. *Computers & Industrial Engineering*.

[Giret and Botti, 2004] Giret, A. and Botti, V. (2004). Holons and agents.

[INDIA BRAND EQUITY FOUNDATION, 2015] INDIA BRAND EQUITY FOUNDATION (2015). Role of Manufacturing in Employment Generation.

[Jeschke et al., 2017] Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., and Eschert, T. (2017). *Industrial Internet of Things and Cyber Manufacturing Systems*, pages 3–19. Springer International Publishing, Cham.

[Lee, 2008] Lee, E. A. (2008). Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369.

[Li et al., 2018] Li, H., Salvador Palau, A., and Kumar Parlikad, A. (2018). A Social Network of Collaborating Industrial Assets. *Submitted to: Journal of Risk and Reliability*.

[Li et al., 2015] Li, S., Xu, L. D., and Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259.

[Martinsson, 2016] Martinsson, E. (2016). WTTE-RNN : Weibull Time To Event Recurrent Neural Network. *UNIVERSITY OF GOTHENBURG*, (Department of Computer Science and Engineering).

[Mönch and Drießel, 2005] Mönch, L. and Drießel, R. (2005). A distributed shifting bottleneck heuristic for complex job shops. *Computers and Industrial Engineering*.

[Ning and Wang, 2011] Ning, H. and Wang, Z. (2011). Future internet of things architecture: Like mankind neural system or social organization framework? *IEEE Communications Letters*, 15(4):461–463.

[Panait and Luke, 2005] Panait, L. and Luke, S. (2005). Cooperative multi-agent learning: The state of the art survey. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434.

[Salvador Palau, 2017] Salvador Palau, A. (2017). *Developing Collaborative Learning in Prognostics-First year report*. PhD thesis, University of Cambridge.

[Saxena and Goebel, 2008] Saxena, A. and Goebel, K. (2008). Turbofan Engine Degradation Simulation Data Set. *NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA*, NASA Ames.

[Unland, 2015] Unland, R. (2015). Industrial Agents. In *Industrial Agents*, pages 23–44. Elsevier.

[Vijayaraghavan et al., 2008] Vijayaraghavan, A., Sobel, W., Fox, A., Dornfeld, D., and Warndorf, P. (2008). Improving Machine Tool Interoperability Using Standardized Interface Protocols : MT Connect. In *2008 International Symposium and Flexible Automation,*, pages 21–24.

[Xu et al., 2014] Xu, L. D., He, W., and Li, S. (2014). Internet of things in industries: A survey.

[Yoon et al., 2014] Yoon, J., He, D., and Van Hecke, B. (2014). A PHM approach to additive manufacturing equipment health monitoring, fault diagnosis, and quality control. *PHM 2014 - Proceedings of the Annual Conference of the Prognostics and Health Management Society 2014.*