MODELLING SUPPLY CHAIN QUALITY MANAGEMENT MARKOV CHAIN AND REINFORCEMENT LEARNING

ANNAPOORNI MANI

DOCTOR OF PHILOSOPHY
UNIVERSITI KUALA LUMPUR
2023

MODELLING SUPPLY CHAIN QUALITY MANAGEMENT USING MARKOV CHAIN AND REINFORCEMENT LEARNING

ANNAPOORNI MANI

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY (ELECTRICAL & ELECTRONICS)

UNIVERSITI KUALA LUMPUR BRITISH MALAYSIAN INSTITUTE

January, 2023

TABLE OF CONTENTS

CHAPTER 1

CHAPTER ONE: INTRODUCTION

1.1 PREAMBLE

This introductory chapter presents an overview of the supply chain quality management. The automotive industry is the world's most massive single manufacturing activity and the world's most important economic sectors by revenue. This industry makes use of 15% of the world's steel, 40% of the world's rubber, 25% of the world's glass, and 40% of the annual oil production of the world. Ever since the origin of the automotive industry in the 1890s, there are hundreds of manufactures originated. China and USA top the automotive industry in terms of production, while Japan, Germany, South Korea, India, Mexico, etc., also produce high volumes. The changing expectations of consumers are a direct result of the increased availability of goods and services. The advent of the 'connected car,' driverless cars, and enhanced driver support, as well as improved fuel efficiency and new or improved powertrains, are just a few examples of the ways in which new technologies are radically altering vehicles.

The supply chain for automobiles is accustomed to having to follow manufacturers as they keep expanding their footprints across the world. While meeting the demand from manufacturers and their customers for cheaper and more efficient components and modules, suppliers are required to ensure compliance with newly enacted regulations as well as standards that are becoming increasingly stringent.

As a direct consequence of the commercial and rational significance of this factor, the original equipment manufacturers (OEMs) have begun implementing cost-cutting programmes, which has led to the supply chain being subject to constant cost pressure. The demand from customers for vehicles that are better, cleaner, and more affordable is putting upward pressure on prices all along the supply chain.

In order to accommodate these shifts, the supply chain for automobiles has begun to undergo reorganisation. Larger suppliers have been successful in meeting increasingly complex technological requirements. Small sellers simply do not have the financial resources to invest in transition.

The procurement teams' responsibility to track down all of the affected products and address quality control presents a risk and a challenge for them. If manufacturers maintain their practise of using short development cycles and lengthy and complicated supply chains, it is likely that problems will continue to arise. Countries compete with one another to design and manufacture vehicles that are superior, cleaner, and more affordable in order to meet the demands of both established and emerging markets. It's been fifty years since the automotive supply chain industry had access to skilled labourers who could be employed there. It is very challenging to attract new talents into this sector with knowledge of the new process and new technologies with increasing production demands.

1.2 OVERVIEW OF SUPPLY CHAIN QUALITY MANAGEMENT

Supply chain quality management, also known as SCQM, refers to the process of ensuring that customers receive the highest possible standard of goods or services by means of cooperative quality management of the supply chain on the part of its constituents, such as the buyer and the supplier. SCQM is a systems-based approach to performance improvement that capitalises on opportunities made available by upstream and downstream linkages with suppliers and customers. This definition describes SCQM as "supplier and customer quality management." In other words, in order for businesses to gain a competitive advantage and improve their performance, they need to monitor the efficiency of their internal operations while simultaneously managing the businesses that are a part of their chain or network. Finding and fixing manufacturing flaws is only one aspect of quality management. The goal of quality management is to achieve quality throughout an entire organisation and its supply chain network by exerting an influence on the individual companies that make up the chain.

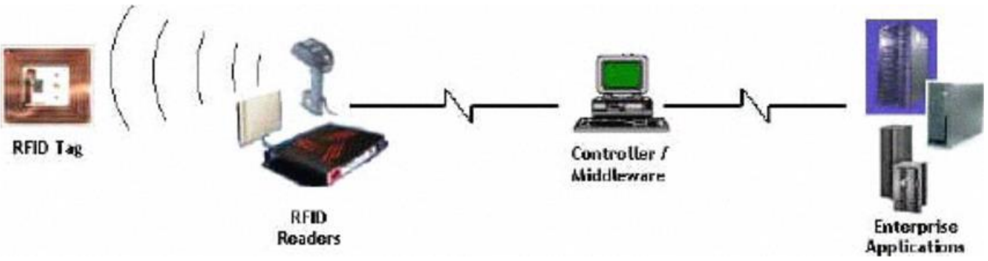


Figure 1.1: RFID system components(al Kattan & AI-Khudairi, 2007)

The modern production system might benefit greatly from the implementation of radiofrequency identification, also known as RFID. This would allow for real-time data collecting as well as improved control of industrial processes. Up to this point, RFID technology has seen extensive use in the manufacturing industry. Some of the application domains in this industry include the design of products and processes, assembly, planning of materials, quality control, scheduling, and maintenance, amongst others. The majority of RFID applications focus on time. The timestamps that correspond to the occurrence of RFID-based events are related with the events themselves. The events may indicate a variety of semantics, including beginning/ending, moving to a new location, modifying existing relationships, or the occurrence of a new event.

The modelling of the flow of material through the arriving inspection station is the primary focus of this research. In recent years, there has been a growing interest in the application of RFID technology in manufacturing and supply chain operations. In this investigation on the operation of the RFID system, the Markov chain method is utilised. During this research, the automotive supply chain's input inspection section was taken into consideration. In the area designated for the receiving and inspection of raw materials, an RFID system has been installed. There are seven steps that make up the incoming inspection: the Packaging Inspection, the Visual Inspection, the Gauge inspection, the Pack and store stage, and the Rework and Return stage. Throughout each of the six stages of the inspection, RFID data is collected and then mapped accordingly. The movement of components is illustrated by a state machine diagram. In order to model the behaviour of the RFID Markov chain during the incoming inspection of the raw materials, a model has been developed. There is an expectation that the materials provided by the Tier 1 supplier will conform to the specifications stated by the OEM. The materials and parts that are able to meet the criteria that have been established are accepted and kept in the warehouse.

In the meantime, the items that have only small deviations from the standard are reworked and accepted once the inspection has been qualified. The items with serious nonconformities or those that cannot be repaired are sent back to the vendor for rejection. A finite state diagram is used to illustrate the flow of the process, and weights are allotted in accordance with that diagram. The Markov chain model that was presented is used to make an estimate of the transition probability of the materials, and then the results of this estimation are explained. Utilizing this model can be of assistance in the construction of a reliable and secure automotive supply chain.

1.3 MOTIVATION

Research motivation on reinforcement learning in Quality 4.0 using active learning can include the following:

1. Improving the efficiency and reducing costs of manufacturing processes: By using RL and active learning, manufacturing processes can be optimized to reduce waste, improve production rates, and lower costs.
2. Enhancing the quality of products and services: By using RL and active learning, the quality of products and services can be improved by identifying and addressing defects or issues early on in the manufacturing process.
3. Adapting to changing conditions: RL and active learning can allow a system to adapt to changes in the manufacturing process, such as changes in demand or supply, and maintain optimal performance.
4. Handling large and complex data: RL and active learning can handle large and complex data from the Internet of Things (IoT) and other Industry 4.0 technologies, such as sensors and machine learning, to improve the decision-making process in manufacturing.
5. Human-in-the-loop decision making: RL and active learning can enable human-in-the-loop decision making in manufacturing by integrating human expertise and feedback into the decision-making process.
6. Personalization: RL and active learning can enable personalization in manufacturing by enabling the system to adapt to individual preferences and needs.
7. Handling uncertainty: RL and active learning can handle uncertainty in manufacturing processes by allowing the system to make decisions based on uncertain or incomplete information.
8. Continual learning: RL and active learning can enable the system to continue learning and adapt to new information and changing conditions over time.
9. Robustness: RL and active learning can enable the system to function robustly in the presence of noise or disturbances in the manufacturing process.

10. Scalability: RL and active learning can scale to handle large and complex manufacturing systems and environments.

1.3.1 INTERNATIONAL STANDARDS (ISO AND IATF)

International Standards Organization (ISO) proposed ISO 9001:2015 guidelines for Quality management systems.

When a business meets the criteria outlined in ISO 9001:2015, it is considered to have a quality management system.

1. it must demonstrate that it is capable of consistently providing products and services that fulfil the standards imposed by customers as well as any applicable legislative and regulatory requirements; and
2. seeks to increase customer satisfaction through the efficient use of the system, which includes mechanisms for the system's continual development and the guarantee of adherence to client needs as well as applicable legislative and regulatory requirements when applicable.
3. The standards of ISO 9001:2015 are all general in nature, and they are designed to be relevant to any company, regardless of the goods and services it offers, its size, or the kind of business it is.
4. International Automobile Task Force (IATF) is an organisation made up of automotive manufacturers and the trade associations that represent each of them. Its purpose is to offer higher-quality goods to clients of automotive companies all over the world.
5. IATF 16949:2016 is the standard proposed by IATF and approved by ISO. This standards guidelines document serves as a standard need for quality management systems in the automotive industry, and it is based on ISO 9001 as well as customer-specific demands from the automotive sector.
6. IATF 16949 places an emphasis on the establishment of a process-oriented quality management system that allows for the reduction of variation and waste in the supply chain, as well as continuous improvement, defect prevention, and provision for the elimination of defects. The objective is to satisfy the requirements of the client in an effective and efficient manner.

Gruszka and Misztal (Gruszka & Misztal, 2017) in their review of the standard highlighted that, the standard encourages automobile manufacturers and their suppliers to invest a significant amount of effort into the quest for management support systems and procedures that are more efficient. To accomplish this, they make use of information and ERP systems that are mediated by IT.

1.3. INDUSTRY 4.0 AND QUALITY 4.0

Industry 4.0, often called as the fourth industrial revolution, integrates sophisticated technologies including IoT, big data analytics, and robotics. into manufacturing and production processes.

Quality 4.0 is an extension of Industry 4.0, with a focus on using these advanced technologies to improve quality management in manufacturing and production. Quality 4.0 aims to improve quality by using real-time data from sensors and other sources to optimize quality control and monitoring, as well as using data analytics and other advanced technologies to predict and prevent defects. Quality 4.0 also includes a focus on the integration of advanced technologies in the entire quality management system such as:

1. Predictive maintenance
2. Real-time Quality control
3. Automation of Quality processes
4. Traceability
5. Cybersecurity

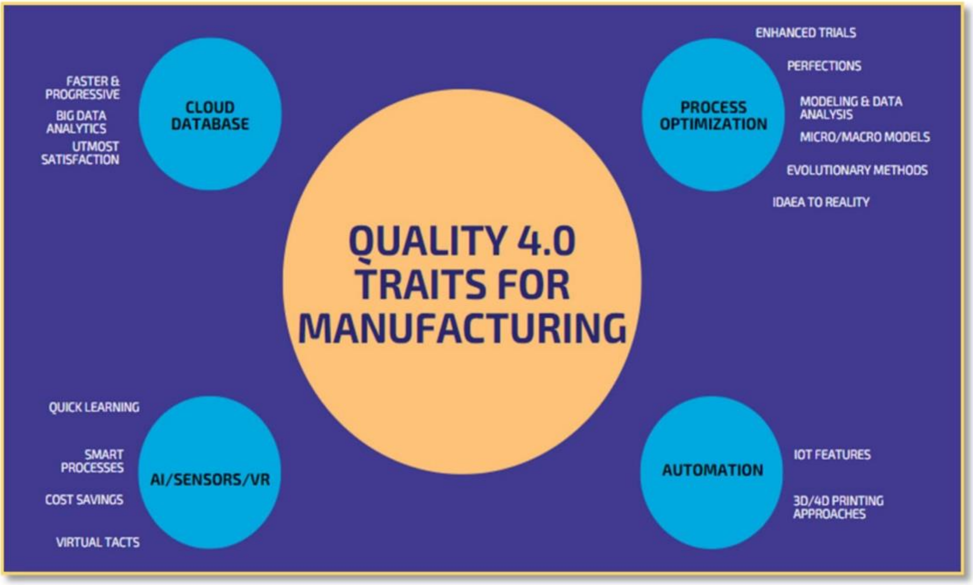


Figure 1.2: Quality traits for Manufacturing(Javaid et al., 2021)

Quality 4.0 can bring several benefits to the manufacturing and production process, such as:

1. Improved efficiency and productivity
2. Increased product quality and reliability
3. Reduced costs.
4. Increased flexibility and adaptability
5. Improved traceability and transparency
6. Better customer satisfaction

It is important to note that, while Industry 4.0 and Quality 4.0 share many similarities, Quality 4.0 places a greater emphasis on the use of advanced technologies to improve quality management and to meet the needs of customers. Therefore, it is important to have a clear understanding of the specific requirements and benefits of Quality 4.0 when implementing Industry 4.0 in the automotive industry (Forero & Sisodia, 2020). The implementation of Industry 4.0 technologies in supply chain quality management in the automotive industry can pose several challenges in terms of compliance with the IATF 16949:2016 standard.

Some of these challenges include:

a) Data management: Industry 4.0 technologies generate large amounts of data, and the management of this data can be challenging. This can make it difficult to meet the requirements for documentation and records defined in the IATF 16949:2016 standard.

b) Traceability: The traceability of products and parts is a key requirement of the IATF 16949:2016 standard. Industry 4.0 technologies can make it more challenging to ensure traceability, especially when using technologies such as 3D printing and robotics.

c) Cybersecurity: Industry 4.0 technologies rely on data networks and cloud-based systems, which can be vulnerable to cyberattacks. Ensuring the cybersecurity of these systems can be challenging, and this can make it difficult to meet the demands of the IATF 16949:2016 standard.

d) Human factors: Industry 4.0 technologies can lead to changes in the roles and responsibilities of employees, and this can make it challenging to ensure the competence and training of employees as per the standard.

e) Auditing and certification: Auditing and certifying Industry 4.0 systems can be challenging, as the systems are often highly complex, and the auditors may not have the necessary expertise.

f) Risk management: Industry 4.0 technologies can introduce new risks that need to be identified, evaluated and controlled as per the standard.

In conclusion, while Industry 4.0 technologies have the potential to significantly enhance supply chain quality management in the automotive industry, their implementation can pose several challenges in terms of compliance with the IATF 16949:2016 standard. Addressing these challenges will be essential for ensuring compliance with the standard.

1.4 PROBLEM STATEMENT

Implementing incoming product quality prediction in Quality 4.0 can be challenging and may face several issues, including: Data availability and quality: Collecting and cleaning large amounts of data from various sources can be difficult and time-consuming. Incorporating human expertise into the decision-making process can be challenging, as it may require a change in the traditional process. Scaling the system to handle large amounts of data and real-time predictions can also be challenging.

The developed prediction model needs to be optimized. Optimizing prediction models in Quality 4.0 can help reduce the number of false positives or negatives in the manufacturing process. Optimizing the models can improve their performance and scalability when dealing with large and complex data. The manufacturing process may have a high degree of uncertainty, such as variations in the quality of incoming products. Manufacturing process may change over time and the models must adapt to these changes.

Automated visual inspections (AVI) use computer vision and machine learning techniques to automate the process of inspecting products for defects or other issues. One of the main difficulties in implementing AVI is the need for high-quality labelled data to train the models. The difficulties in operations due to labelling data can include. Scaling the data collection and annotating process to handle large amounts of data can be difficult and may require significant resources.

1.4.1 CHALLENGES IN IMPLEMENTING INDUSTRY 4.0 IN AUTOMOTIVE QUALITY MANAGEMENT

There are several hurdles that can make it difficult to implement Industry 4.0 technologies in automotive quality management. Some of the main hurdles include:

a) High initial investment: Implementing Industry 4.0 technologies can be costly, and many companies may struggle to allocate the necessary budget and resources.

b) Limited technical expertise: Many companies lack the technical expertise needed to implement Industry 4.0 technologies, which can make it difficult to fully utilize the potential of these technologies.

c) Limited understanding of the potential benefits: Many companies lack a clear understanding of the potential benefits of Industry 4.0 technologies, which can make it difficult to justify the investment.

d) Limited integration with existing systems: Many companies struggle to integrate Industry 4.0 technologies with their existing systems, such as ERP and MES systems, which can make it difficult to fully utilize the potential of these technologies.

e) Cybersecurity concerns: Cybersecurity concerns can also be a barrier to implementation, as Industry 4.0 technologies often rely on data networks and cloud-based systems, which can be vulnerable to cyberattacks.

f) Limited collaboration: Collaboration between different departments and stakeholders is often limited, which can make it difficult to effectively implement Industry 4.0 technologies and to measure their impact on supply chain quality management.

g) Limited standardization: There is a lack of standardization in the implementation of Industry 4.0 technologies, making it difficult for companies to effectively integrate these technologies into their supply chain.

h) Data management and Quality: Industry 4.0 technologies generate large amounts of data, and the management of this data can be challenging, which can make it difficult to meet the requirements for documentation and records defined in the IATF 16949:2016 standard. Ensuring the quality of data can be challenging with the new data sources.

Addressing these hurdles will be essential for successfully implementing Industry 4.0 technologies in automotive quality management and realizing the potential benefits such as improved efficiency and quality.

1.5 RESEARCH OBJECTIVES

The aim of this research work is to investigate the problems associated with supply chain quality and to develop optimization algorithms. The research focuses on the optimization of the supply chain quality by improving the process and adapting to industry 4.0 and quality 4.0 guidelines. To following objectives are formulated based on the problems highlighted in the previous section of this chapter.

1. To develop a prediction model for Quality 4.0 using Markov Chains.
2. To optimize the developed Markov Decision Process models through reinforcement learning algorithms.
3. To improve the performance of the automated visual inspection models using deep active learning approach.

1.6 SCOPE OF THE THESIS

The scope of the thesis is limited to developing a model. An incoming inspection section in the automotive supply chain inventory is considered for this research work. Six states in the incoming inspection, namely packaging inspection, visual inspection, gauge inspection, rework, return and pack and store are considered in this research work. The flow of the material from packaging inspection either to pack and store section or return section is represented using a finite state machine (FSM). A Markov Chain (MC) model is developed, and the trajectory is estimated. Three weight values are considered throughout the research work. To further enhance the performance of the model, and the optimization algorithm is developed using reinforcement learning. The validity of the model is tested using simulated RFID data.

The procurement teams' responsibility to track down all of the affected products and address quality control presents a risk and a challenge for them. If manufacturers maintain their practise of using short development cycles and lengthy and complicated supply chains, it is likely that problems will continue to arise. Countries compete with one another to design and manufacture vehicles that are superior, cleaner, and more affordable in order to meet the demands of both established and emerging markets. It's been fifty years since the automotive supply chain industry had access to skilled labourers who could be employed there. It is very challenging to attract new talents into this sector with knowledge of the new process and new technologies with increasing production demands.

1.7 ORGANIZATION OF THE THESIS

This thesis encapsulates the supply chain quality management, and the model developed using the Markov chain model. Reinforcement learning is used to optimize the developed model. The performance of the model is validated using the simulated RFID dataset. The research focussed on improving the supply chain quality, thereby increasing the productivity and quality of the process in the supply chain. This thesis is documented in separate chapters contributing to the objectives of this research.

Chapter 1 introduces supply chain quality management. The motivation for this research is described in the motivation section. The core problems leading to this research are pointed out in the problem statement section. The objectives of this research are listed in the objective section, whereas the scope and limitation of this research are documented in the research scope section. The section-wise detailing of the chapters is presented in the thesis organization section.

Chapter 2 generally presents the theory and literature related to the key areas focussed on this research work: Quality inspection, Markov chain, reinforcement learning and deep active learning. The chapter is subdivided into appropriate sections. Section 2.2

discusses the Quality management and narrows down to supply chain quality and incoming inspection. This section also reviews the state-of-the-art literature related to supply chain quality management. The issues present in the traditional supply chain is then highlighted. Section 2.3 details about the relationship between markov chain and manufacturing and further explains the fundamental concepts of markov chains. The basics of the markov chain properties are eloborated. The literature is then reviewed, and the findings are highlighted. Section 2.4 explains about the connection between markov decision process and basics concepts of reinforcement learning and its algorithms. The section further explains the use of reinforcment learning to solve the markov decision process. The state of the art literature is studied in detail and the research gap is highlighted. Section 2.5 details the fundamentals of the automated quality inspection, the needs for active learning and eloborates further on the deep active learning approach. The literature and research gaps are further highlighted in Section 2.6. Finally the topics presented in this chapter are summarized in Section 2.7 Chapter 3 describes the research methodology in detail in three segments. Section 3.2 introduces the Markov decision process, and the incoming inspection case study using the RFID tag data. The state machine diagram is formulated to explain the case study. The incoming inspection case study and the substations involved are described in detail. The section further demonstrates the computation of Markov chain properties for the experimental scenarios. The section additionally describes the transition matrix and transition probability estimation and the steps used for the development of the raw material acceptance prediction model. Section 3.3 explains the algorithm development using reinforcement learning to optimize the prediction model developed using Markov decision process solved in the above sections. This section subsequently details the steps to solve the raw material acceptance prediction model based on the dynamic programming and temporal difference RL algorithms. Likewise, the section explains the procedure for the model development for the estimation of the material quality categorization using temporal difference algorithm. Section 3.4 outlines the impeller defects casting dataset and details the traditional deep learning approach for visual inspection. The section further details the steps involved to optimize the model development by optimal usage of data based on deep active learning approach. Lastly, section 3.5 summarizes the methods described in this chapter and directs to the results chapter.

Chapter 4 marks the results of the optimization models and discusses the results with the literature and finally summarizes the findings. The chapter is grouped into three sections. Section 4.2 demonstrates the results of the raw material acceptance prediction model for the incoming inspection case study using Markov decision process. The results are then discussed along the Markov properties and compared with the literature. Section 4.3 exhibits the results for the optimized raw material acceptance models using dynamic programming and temporal difference. The results are compared and then discussed in detail. Further, the results of the material quality estimation model are presented and discussed accordingly. Section 4.4 showcases the results of visual inspection model using deep learning computer vision models. The results based on the traditional deep learning approach and the deep active learning approach are elaborated and analysed. In the end, section 4.5 summarizes the results presented in the chapter.

Chapter 5 summarizes the objectives of the thesis and the overall research findings made in this thesis. The chapter further presents the recommendations and improvement for future research.

CHAPTER 2

CHAPTER TWO: LITERATURE REVIEW

2.1 CHAPTER INTRODUCTION

In Chapter 2, the theory and literature linked to the important areas that this study effort is focusing on are presented in a broad sense. These key topics are quality inspection, the Markov chain, reinforcement learning, and deep active learning. The chapter is broken up into the relevant portions that make sense. In Section 2.2, we talk about quality management, specifically focusing on the quality of the supply chain and the entering inspection. In this section, we will also take a look at the most recent research that has been done in the field of supply chain quality management. The problems that are inherent in the conventional supply chain are then brought to light. In section 2.3, we go into the specifics of the connection between a Markov chain and the manufacturing process, in addition to expanding on the underlying ideas of Markov chains. The fundamental aspects of the Markov chain attributes are broken out in detail. After that, a review of the previous research is conducted, and its findings are emphasised. In section 2.4, the relationship between the Markov decision process and the fundamental ideas behind reinforcement learning and its algorithms is broken down and explained. The following explanation delves deeper into the application of reinforcement learning as a solution to the Markov decision process. The most recent research is critically evaluated, and any knowledge gaps that remain in the field are pointed out. In Section 2.5, the principles of the automated quality inspection are broken down, as are the requirements for active learning, and further information is provided regarding the deep active learning strategy. In Section 2.6, the gaps in the study and the literature are discussed in further detail. At long last, a synopsis of the material covered in this chapter may be found in Section 2.7.

2.2 QUALITY MANAGEMENT

Supply chain management (SCM) is the heart of any business process and becomes the survival factor for the automotive sector. Managing a supply chain is a cumbersome task for any industry. Logistic delays, shipping damages, poor or less quality checks due to mass production, failure to deliver shipments on time are some of the known risks and challenges of the supply chain. With no international standards devised, automotive manufactures follow in-house standards to manage supply chain. The production downtimes are unavoidable as the market requirements keeps changing. A well-managed material requirement forecasting can certainly avoid the material shortage for smooth production. Manufacturers are hesitant to invest in adopting new technological changes. Due to this reluctant nature, industries pay high cost for the people with knowledge of obsolete technologies.

The automotive industry is the most significant economic sector in the world in terms of revenue, and it is also the single most important manufacturing activity in the world. This industry accounts for 15% of the world's steel production, 40% of the world's rubber production, 25% of the world's glass production, and 40% of the world's annual oil production. Ever since the origin of the automotive industry in 1890s, there are hundreds of manufactures originated. China and USA top the automotive industry in terms of production, while Japan, Germany, South Korea, India, Mexico etc., also produces high volumes. The changing expectations of consumers are a direct result of the greater availability of goods and services. The emergence of the 'connected automobile,' autonomous cars, and greater driver support, as well as higher fuel efficiency and new or upgraded powertrains, are just a few examples of the ways in which new technologies are radically altering vehicles. The automotive supply chain is accustomed to following manufacturers as they continue to grow global footprints. Suppliers must ensure compliance with new regulations and increasingly demanding standards, while meeting demand from manufacturers and their customers for cheaper, more efficient components and modules.

The OEM has implemented cost-cutting programmes because of the commercial and logical significance, putting constant cost pressure on the supply chain. Consumers' demands for safer, more environmentally friendly vehicles at lower prices are increasing the amount of stress placed on the automotive industry's supply chain.

With these shifts in mind, the automotive supply chain has begun reorganising. In spite of the growing complexity of the technological requirements, larger suppliers have been able to keep up. Fewer suppliers can afford the necessary investments to adapt.

Procurement teams facing the risk and complexity of tracing all affected products and addressing quality control. If production facilities keep using lengthy and complicated supply chains and a rapid pace of innovation, similar issues will arise again and again. There is international competition to produce automobiles that meet the needs of both developed and developing economies at the lowest possible cost. Approximately half of the skilled workers currently employed in the automotive supply chain have reached retirement age. With rising production demands, it is difficult to entice new talent into the industry who is familiar with the latest processes and technologies.

Raw materials in stock are those that have been procured from vendors and will serve as process inputs. Reducing production downtime, raw material management ensures the right amount of raw materials are stored in the right place at the right time for the lowest possible price. Material flow can be linked with production, allowing for minimal costs to be incurred.

2.2.1 SUPPLY CHAIN QUALITY MANAGEMENT

In the inventory, the term "raw materials" refers to any components or supplies that were initially procured from external suppliers before being incorporated into the production procedure in some capacity. Raw material management helps cut down on production downtime by ensuring that the necessary amount of material is stored in the appropriate location within the raw material inventory at the appropriate time at a cost that is affordable. By exercising effective management, these costs can be kept as low as is practically possible, and the flow of materials can be synchronised with production (Tersine, 1994)

Keeping a competitive advantage, remaining in business, and expanding their market share are all goals that can be accomplished through effective inventory management. It has been calculated that proper management of inventories accounts for between 30-35 percent of the total material value (Wallin et al., 2006).

According to Taiichi Ohno, one of the seven wastes that ought to be avoided is inventory. The most obvious part of a company's overall assets, it only makes up between 5-30 percent of the entire assets (Goldsby & Martichenko, 2014). It is essential to design a model that satisfies the diverse material management requirements of businesses in a way that is efficient with their financial resources.

In lean manufacturing, a defect is one of the wastes that is among the most significant. Both a flawed production process and faulty components in the raw materials used to make the product can contribute to quality issues. By resolving flaws in the raw materials throughout the manufacturing process, it may be possible to shorten the takt time of the production line and enable lean manufacturing.

2.2.2 INCOMING INSPECTION

Incoming inspection is a quality control process used by automotive OEMs (original equipment manufacturers) to ensure that the parts and components they receive from suppliers meet their specified requirements. The process involves inspecting and testing the incoming materials to verify that they are of the correct quality and meet the required specifications.

The specific operations involved in incoming inspection can vary depending on the type of parts or components being inspected. However, some common steps in the process include:

1. Receiving and unpacking the materials: The materials are received and checked against the purchase order to ensure that they are the correct items and that they have been shipped in good condition.
2. Visual inspection: The parts or components are visually inspected to ensure that they meet the specified requirements in terms of size, shape, and appearance.
3. Dimensional inspection: The parts or components are measured to ensure that they meet the required dimensional tolerances.
4. Functional testing: The parts or components are tested to ensure that they function correctly and meet the required performance specifications.
5. Sampling and testing: A sample of the materials is selected for further testing to ensure that they meet the required quality standards.
6. Documenting and reporting: The results of the inspection are documented and reported to the relevant parties. If any non-conformances are found, appropriate corrective actions are taken.

The goal of incoming inspection is to identify and reject any non-conforming parts or components before they are used in the manufacturing process, to maintain the quality of the final product and avoid costly rework or scrap.

The inbound inspection technique is part of quality endurance, and its purpose is to ensure that the quality of the raw materials satisfies the criteria that have been mutually agreed upon by the customer and the seller. In modern times, this human-centric procedure has been mechanised by deploying numerous competitive technologies, with the deployment of Radio Frequency Identification (RFID) being one of the first of these technologies. Within the realm of automated identification systems, radio frequency identification tags represent a major step forward. Intelligent bar codes, also known as radio frequency identification (RFID) tags, contain a chip that can capture data pertaining to the item being scanned. These tags are read by an RFID reader, which subsequently makes it possible to follow the movement of things. This technology was initially developed to keep track of cattle, but its uses have since expanded to include tracking vehicles (Weinstein, 2005), pets (Rieback et al., 2006), and things in the manufacturing process (Sharpe et al., 2012). Cattle were the original focus of this technology's development. It is possible to arrive at incorrect conclusions through the modelling of manufacturing processes that involve humans. In the context of an investigation into a highly manual process, the research conducted by Baines and his team (Baines et al., 2004) investigated ways to improve the accuracy of discrete event modelling.

The data log from these RFID readers is used by decision models, which helps them better grasp the process parameters. One example of this kind of model is called the Markov Decision Process.

The origin of Markov Chains dates to the beginning of 20th Century when Markov chains were first proposed by Andrei Andreevich Markov. His studies about the chains of linked probabilities published in 1906 led to the future of theory and to the basis to several research till date. Markov chains are widely applied in operations research (Winston & Goldberg, 2004). The modelling of stochastic processes using Markov chains and Markov processes falls within the categories of discrete-time and continuous-time processes, respectively. They provide a mathematical representation of a process by illustrating the likelihood of it transitioning between phases. The purpose of this study was to help in achieving right-first-time manufacturing. Within the scope of this work was the connection between discrete event simulations and human performance models. Each station is equipped with RFID readers, and the raw materials that move between the stations are tagged with RFID.

Batson and McGrough (Batson & McGough, 2007) presented a new direction to the quality inspection in SQCM.

Jiaqi (Jiaqi, 2010) discussed the modelling approach Quality inspection in SQCM.

2.3 MARKOV CHAINS AND MANUFACTURING

Manufacturing Matters! In fact, the wealth of a nation may either be extracted from the earth (using natural resources and agriculture) or created (via the addition of value to materials through processing). As a result, the importance of the manufacturing industry cannot be overstated because it is one of only two means to generate national wealth.

1. Chemical, materials, and electricity: continuous manufacturing because their fundamental processes develop in a continuous manner throughout time.
2. Automotive, electronics, appliance, aerospace, and other markets: manufacture discrete parts.

Both continuous and discrete production methods can be utilised in the manufacturing process (J. Li & Meerkov, 2009).

2.3.1 MARKOV CHAIN PROPERTIES

The mathematical system known as a Markov chain change from one state to another in accordance with predetermined probabilistic rules. Whatever the initial conditions were that led to the current state of the system, the set of possible future states is fixed in a Markov chain. That is to say, given a specific initial condition and enough time has passed, any possible future state can be predicted with certainty. A few characteristics of Markov chains are as follows:

1. It's essential for a Markov chain to have a finite number of states, so the state space is finite. Seven discrete states are considered in this case.
2. Second, it has no long-term memory, so the probability of changing states depends only on the current one and the amount of time that has passed. The past has no bearing on the present or the future.
3. Third, irreducibility means that any state can be reached from any other state by taking an infinite number of transitions.

4. Fourth, the chain may return to the same state after a fixed number of steps if the state is periodic.
5. Fifth, the chain is aperiodic if and only if it is not periodic. As a result, it is possible to revert to a previous state, but this may not occur after a predetermined amount of time has passed.

When given enough time, a Markov chain will reach every possible state with probability 1. This property is known as ergodicity. Because of this, we can say that the sequence is nonreducible and aperiodic.

2.3.1.1 TRANSITION MATRIX

In Markov chain process, a transition matrix is a mathematical representation of the probabilities of transitioning from one state to another. In the context of automotive quality management, a transition matrix could be used to model the probability of a vehicle or component transitioning from one state to another, such as from "in production" to "inspection" or "failure."

$$\mathbf{T} = \begin{pmatrix} p_{00} & p_{01} & \cdots & p_{0j} & \cdots & p_{0n} \\ p_{10} & p_{11} & \cdots & p_{1j} & \cdots & p_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdot\cdot & \cdots \\ p_{i0} & p_{i1} & \cdots & p_{ij} & \cdots & p_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdot\cdot & \cdots \\ p_{n0} & p_{n1} & \cdots & p_{nj} & \cdots & p_{nn} \end{pmatrix} \qquad (1)$$

Figure 2.1: A typical Transition Matrix

The transition matrix is a square matrix, as shown in Figure 2.3, where the rows and columns represent the different states in the system, and the entries in the matrix represent the probability of transitioning from one state to another. For example, in a manufacturing process, the states could be "in production," "inspection," "rework," and "failure," and the matrix would have four rows and four columns, with entries representing the probability of transitioning from one state to another.

By analysing the transition matrix, it is possible to identify patterns and areas for improvement in the production process. For example, a high probability of transitioning from "inspection" to "rework" or "failure" could indicate a problem with the inspection process, while a high probability of transitioning from "in production" to "failure" could indicate a problem with the production process.

Additionally, by using the transition matrix and the current state of the system, it's possible to predict the future state of the system and plan accordingly. By analysing the transition matrix, it's also possible to identify the steady state, which is the state where the system is going to spend most of the time.

In summary, the Transition matrix is a powerful tool in Markov Chain process, it can be used in Automotive quality management to model the probability of a vehicle or component transitioning from one state to another, identify patterns and areas for improvement in the production process, predict the future state of the system and plan accordingly and identify the steady state.

2.3.1.2 N-STEP TRANSITION PROPERTY

In a Markov chain, the n-step transition probability gives the probability of transitioning from one state to another after n steps. It is the probability of being in a particular state after n steps, given that the current state is known.

The N-step transition probability of a Markov chain is the probability of transitioning from one state to another after N time steps. It is related to the one-step transition probability (also known as the transition probability or transition matrix) of the chain, denoted by P, and is defined as follows:

$P^N [i, j]$ = probability of transitioning from state i to state j after N time steps (2)

where P^N [i, j] is the N-step transition probability from state i to state j, and P [i, j] is the one-step transition probability from state i to state j.

For example, if P is the transition matrix of a Markov chain, then P^2[i, j] is the probability of transitioning from state i to state j in two-time steps, and P^3[i, j] is the probability of transitioning from state i to state j in three-time steps.

The n-step transition probability can also be used to find the probability of certain sequences of states occurring in the Markov chain. For example, if you know the probability of transitioning from state ‘i' to state ‘j’ in one step, and the probability of transitioning from state ‘j’ to state ‘k’ in one step, you can use the 2-step transition probability to find the probability of transitioning from state ‘i’ to state ‘k’ in two steps. This can be useful in analysing the behaviour of the Markov chain over time.

Additionally, n-step transition probability can also be used to calculate the probabilistic measure called the n-step probability distribution, which gives the probability of being in a particular state after n steps. This is also a way of studying the long-term behaviour of the Markov Chain

The n-step transition probability can be computed using the matrix product of the transition matrix raised to the power of n. The entries of the matrix thus obtained give the n-step transition probability.

The n step probability here, is used to answer, “If a Markov chain is in the state I at time t, what is the probability that it will be in state j after n periods.”

The N-step transition probabilities can be used to compute the probability of reaching a particular state after a certain number of time steps, or to analyse the long-term behaviour of the system. They can also be used in reinforcement learning algorithms to estimate the value of states and actions in an MDP.

2.3.1.3 STEADY STATE PROBABILITY

The steady-state probability is the limit of the n-step probability as n goes to infinity. This means that for large values of n, the n-step probability is very close to the steady-state probability, and as n increases, the difference between the two probabilities becomes smaller and smaller.

In a Markov chain, the steady state probability of a state is the probability of being in that state at a given time, if the system has reached a long-term equilibrium. In other words, it is the probability of being in that state when the probability distribution over the states of the system is constant over time.

The steady state probabilities of the states in a Markov chain can be calculated using the stationary distribution, which is a probability distribution that is invariant under the transition probabilities of the chain. In other words, if the current state of the system is given by the stationary distribution, then the probability distribution over the states of the system will remain the same after one time step.

The steady state probabilities can be computed by

1. Linear Equation method
2. Power method

Liner equation method

To find the stationary distribution of a Markov chain, the following steps are followed:

1. Write down the transition matrix of the Markov chain, where each element P [i, j] represents the probability of transitioning from state i to state j.
2. Let π be the stationary distribution, where π[i] is the steady state probability of state i.
3. Set up a system of linear equations by setting the steady state probability of each state to be equal to the sum of the probabilities of transitioning to that state from all other states, as follows:

   $$\pi[i] = \Sigma j\ P\ [j, i]\ \pi[j] \qquad (3)$$

4. Solve the system of linear equations to find the stationary distribution.

Alternatively, you can use the power method to approximate the stationary distribution of a Markov chain.

Power method

This involves starting with an initial probability distribution over the states, and iteratively applying the transition matrix until the probabilities converge to a stationary distribution.

The power method is an iterative algorithm that can be used to approximate the stationary distribution of a Markov chain, which is the long-term probability distribution of the states in the chain. The stationary distribution is a unique probability distribution that is invariant under the dynamics of the Markov chain, and it can be used to determine the long-term behaviour of the chain.

The power method works by iteratively applying the transition matrix of the Markov chain to an initial probability distribution vector. The algorithm starts with an initial probability distribution, and at each step, it multiplies the current probability distribution by the transition matrix of the chain. The process is repeated many times, and the result will converge to the stationary distribution of the Markov chain.

The power method can be summarized as follows:

1. Select an initial probability distribution vector x0, which should be a probability vector, with all non-negative entries, and the sum of entries equal to 1.

2. Multiply the initial probability distribution vector by the transition matrix P of the Markov chain to obtain a new probability distribution vector: x1 = Px0
3. Repeat step 2, by multiplying the current probability distribution vector by the transition matrix P to get the new probability distribution vector: x2 = Px1, x3 = Px2, ...
4. As the number of iterations increases, the sequence of probability vectors {xn} will converge to the stationary distribution vector.

The power method is a simple and efficient algorithm to approximate the stationary distribution of a Markov chain.

2.3.1.4 FUNDAMENTAL MATRIX

The fundamental matrix of a Markov chain is a matrix that describes the long-term behaviour of the chain. It is defined as the matrix of expected number of times that the system will be in each state before it is absorbed, given that it starts in a particular state. Formally, the fundamental matrix for a Markov chain with transition matrix P and absorbing states S is given by:

$$F = (I - Q) ^\wedge (-1) \qquad (4)$$

where I is the identity matrix, and Q is the matrix obtained from P by replacing all the rows corresponding to the absorbing states with zeros.

The elements of the fundamental matrix represent the expected number of times that the system will be in each state before being absorbed, given that it starts in a particular state. The fundamental matrix is useful for analysing the long-term behaviour of a Markov chain, as it tells us the expected number of times that the system will be in each state before being absorbed. It can be used to calculate various performance measures, such as the expected time to absorption and the expected number of transitions between states.

2.3.1.5 LIMITING MATRIX

The limiting matrix of a Markov chain is a matrix that describes the long-term behaviour of the chain in the limit as the number of time steps goes to infinity. It is defined as the matrix of limiting probabilities of being in each state, given that the system starts in a particular state. Formally, the limiting matrix for a Markov chain with transition matrix P is given by:

$$L = PL \tag{5}$$

where L is the matrix of limiting probabilities, and P is the transition matrix of the Markov chain. The elements of the limiting matrix represent the probability of being in each state in the long term, given that the system starts in a particular state.

The limiting matrix is useful for analysing the long-term behaviour of a Markov chain, as it tells us the probability of being in each state in the long term. It can be used to calculate various performance measures, such as the expected number of transitions between states and the expected time spent in each state. The limiting matrix can be computed by iteratively applying the transition matrix P to itself until the probabilities converge to a fixed point. This process is known as finding the stationary distribution of the Markov chain.

2.3.1.6 ABSORBING MATRIX

An absorbing Markov chain is a special type of Markov chain that has at least one absorbing state, which is a state that once entered, cannot be left. This means that once the system reaches an absorbing state, it remains in that state for all future time steps.

Absorbing Markov chains are often used to model systems that have a finite number of possible outcomes, where the system eventually reaches one of these outcomes and remains there. For example, an absorbing Markov chain could be used to model the progression of a disease, where the states represent the different stages of the disease, and the absorbing states represent the end states of "recovered" or "deceased".

One common use of absorbing Markov chains is in the analysis of reliability of systems, where the states represent different levels of system failure, and the absorbing states represent complete failure or repair. Absorbing Markov chains can also be used in economics to model the behaviour of firms, where the states represent different levels of profitability, and the absorbing states represent bankruptcy or success.

2.3.2 RELATED LITERATURE FOR MARKOV CHAINS IN MANUFACTURING

Nalubowa and his associates (Nalubowa et al., 2021) presented a detailed review on the application of Markov chain in manufacturing systems. The literature study outlined the uncertainty modelling in manufacturing systems using Markov chains. They detailed the fundamental concepts of the Markov Chains, Markov properties and classes of Markov chains. Further in their investigation, the authors tabulated the publications in the field of Markov chain for the past two decades. Finally, the authors commented that more research is needed for continuous time Markov chain while there is abundant research carried out in discrete time Markov chains.

Yang (Yang, 2020) investigated the use of Markov Chain properties. Shinoi (Shirai, 2014) highlighted the problem definition and the relation between Markov Chains and reinforcement learning. Jacob (Jacob, 2005) presented the Markov Process and its applications. Leigh et al., (Leigh et al., 2017) developed a Markov chain model that can be used to make predictions about the future. Tochukwu and Hyachinth (Tochukwu & Hyacinth, 2015) developed an agent-based model in which all the input on the model's behavior was represented as a Markov chain. Kiassat, Safaei, and Banjevic (Kiassat et al., 2014) developed a Markov chain technique to anticipate the manufacturing output of a human-machine system while considering HR variables and operator learning. Gingu and Zapciu (Gingu & Zapciu, 2017) provided a solution by eliminating the need for intermediary inventories while maintaining a steady and foreseeable demand for these goods on the market (achieving a balance between supply and demand). Yi and Grossmann (Ye et al., 2019) developed a solution to a continuous time Markov chain model.

2.3.3 RESEARCH GAP IN MARKOV CHAINS

This section details the research gap from the literature and narrows down to the current research topic.

The implementation of Markov chain process in Automotive Quality 4.0 is a relatively new and active area of research. While many studies have shown the potential of Markov chain process in improving the quality of vehicles and components, there are still several research gaps that need to be addressed in order to fully realize its potential in the Automotive Quality 4.0.

One research gap is the integration of Markov chain process with other Quality 4.0 technologies, such as Industry 4.0, Internet of Things (IoT), and Big Data Analytics. While Markov chain process can be used to model and analyze the production process, it can be further enhanced by integrating it with other technologies to provide a more complete and accurate picture of the production process, and to identify areas for improvement.

Another research gap is the scalability and robustness of Markov chain process in large and complex systems. As the automotive industry becomes more complex and more data-driven, it becomes increasingly important to develop Markov chain process models that can handle large and complex systems, and that can be robust to changes and disturbances in the production process.

Another gap is the integration of Markov chain process with real-time monitoring and control systems. Real-time monitoring and control systems allow companies to track the production process in real-time and make adjustments as needed. However, the integration of Markov Chain process with real-time monitoring and control systems is still an open problem.

Additionally, research gaps exist in the development of efficient algorithms for solving large-scale Markov Chain models, as well as the development of methods for validating the results obtained from Markov Chain models.

Overall, there is a significant potential for Markov chain process in Automotive Quality 4.0, but research is needed to fully realize this potential, by addressing the research gaps such as the integration with other Quality 4.0 technologies, scalability and robustness, real-time monitoring and control systems, efficient algorithms and validation methods.

With the research gap and knowledge presented above, our work attempts to solve the problem with quality incoming inspection problem by solving the incoming inspection case study. The first objective of our research focuses on determining the estimation of acceptance probability. The computation of the acceptance probability model is further explained in Chapter 3 of this thesis.

2.4 REINFORCEMENT LEARNING AND MARKOV DECISION PROCESS

Reinforcement learning (RL) is a type of machine learning that is used to train agents to make decisions in an environment by maximizing a reward signal. In a RL environment, rather than instructing the agent in what to do or how to accomplish it, the agent is instead provided with a reward for each activity it successfully completes. The award might be either favourable or unfavourable. After then, the agent will begin carrying out the activities that resulted in it being awarded a reward in a favourable sense. As a result, the process is one of trial and error (Ravichandiran, 2018b).

In the context of automotive quality management, RL algorithms can be used to train quality management agents that can learn to make decisions about how to improve the quality of vehicles or components by maximizing a reward signal, such as reducing the number of vehicles that fail inspection or are sent for rework.

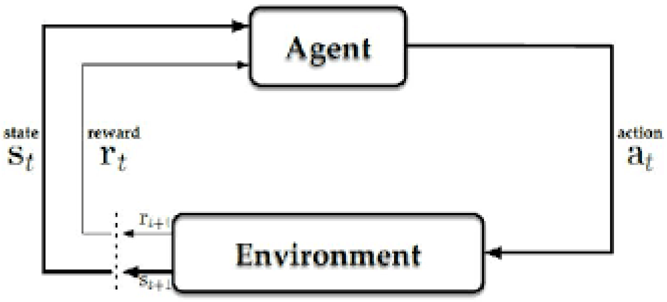


Figure 2.2: The MDP of reinforcement learning.

One application of RL in the automotive industry is in the optimization of production processes. By using RL algorithms, a quality management agent can learn to optimize the production process by experimenting with different actions, such as changing the production schedule or adjusting the parameters of a machine, and observing the effect on the reward signal, such as the number of vehicles that fail inspection.

Another application of RL in the automotive sector could be in the optimization of the supply chain. RL can be used to train an agent that can learn to make decisions about how to optimize the logistics of moving components and vehicles through the supply chain. For example, RL can be used to train an agent to decide the best routes, shipping methods, and suppliers to minimize delays, reduce costs and improve the quality of service.

RL can also be used to optimize the maintenance and repair of vehicles. By using RL, an agent can be trained to make decisions about when to schedule maintenance and repair for vehicles, based on historical data, such as previous maintenance and repair schedules, vehicle usage, and historical failure rates.

RL is a subfield of machine learning that is also sometimes referred to as a semi-supervised learning technique. The use of RL as a framework not only aids in the decision-making process of models but also attracts a lot of attention in game-playing. The RL is analogous to an MDP in that it has a collection of states, a set of actions, transition dynamics, reward functions, and discount factors.

When the MDP is episodic, which means that its states are reset at the conclusion of each episode, the states, actions, and rewards that occur inside an episode generate a trajectory for the policy. The purpose of the RL was to locate the best possible policy that would result in the highest possible expected return from all the states (Arulkumaran et al., 2017; Y. Li, 2017).

2.4.1 BASIC CONCEPTS IN REINFORCEMENT LEARNING

A typical Reinforcement Learning structure is shown in figure 2.3. In reinforcement learning, Markov Decision Process (MDP) is a mathematical framework used to model decision-making problems.

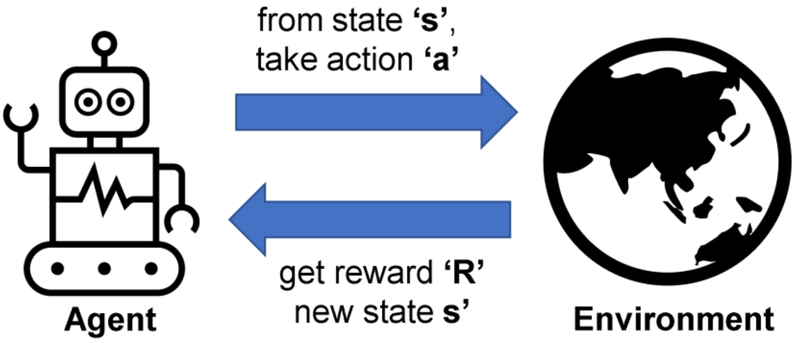


Figure 2.3: Reinforcement learning environment.

The Environment, the Agent, the Action, the State, and the Reward are the core units of RL. An MDP consists of the following components:

States (s): These are the possible situations or configurations of the environment that the agent can be in.

Actions (a): These are the possible choices that the agent can make in each state.

Transition Function: This defines the probability of transitioning from one state to another state based on the action taken. It is denoted as T (s, a, s') and defined as the probability of going from state ‘s’ to state s' by taking action a.

Reward Function: This defines the immediate reward or penalty the agent receives for being in a particular state or taking a particular action. It is denoted as R (s, a) and defines the reward for being in state s and taking action a.

Discount Factor: This is a scalar between 0 and 1 that determines the importance of future rewards relative to current rewards. It is denoted by γ and determines the importance of future rewards over current rewards.

Policy(π): This is the rule or strategy that the agent uses to decide which action to take in each state. It is denoted as π(s) and defines the action to take in state ‘s’.

Value function: This is a measure of how good it is for the agent to be in a particular state or to follow a particular policy. It is denoted as V(s) and it defines the expected cumulative reward for following policy π starting from state ‘s’.

Model: This is a representation of the environment that the agent uses to predict the consequences of its actions. Model-based RL agents use a model of the environment to plan and make decisions, whereas model-free RL agents do not have a model and learn from experience.

To sum up, an MDP in reinforcement learning is a mathematical framework that consists of a set of states, actions, transition function, reward function, discount factor, policy, value function and a model (if needed) that helps the agent to understand and navigate the environment and learn a policy that maximizes the expected cumulative reward over time.

2.4.2 REINFORCEMENT LEARNING ALGORITHMS

Reinforcement Learning Algorithms can be broadly classified into

1) Model based RL algorithms.
2) Model free RL algorithms

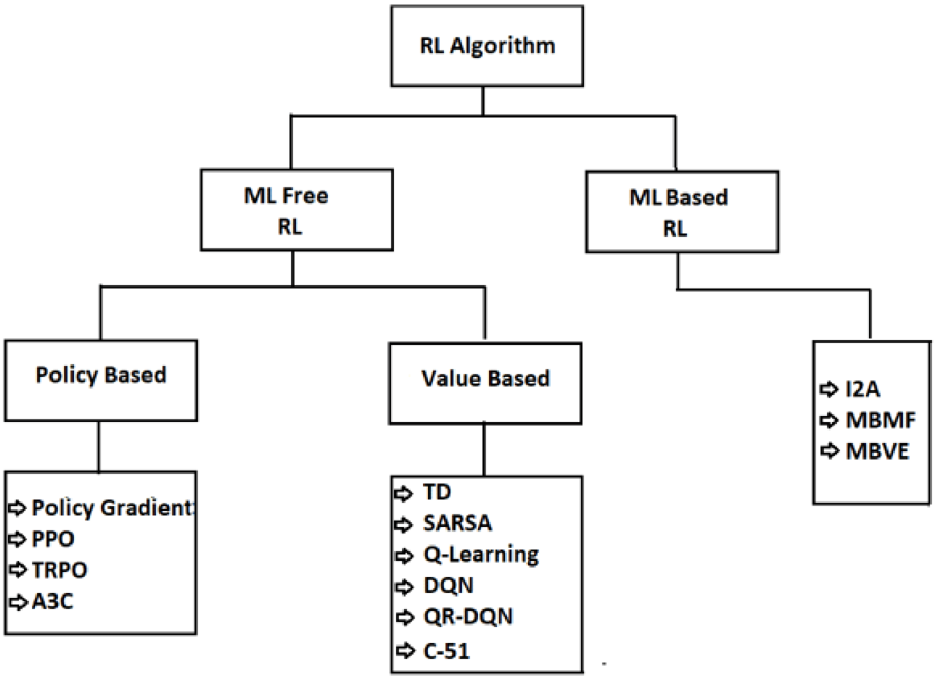


Figure 2.4: Taxonomy of Reinforcement learning algorithms (Akanksha et al., 2021)

2.4.2.1 MODEL BASED REINFORCEMENT LEARNING ALGORITHMS.

Model-based reinforcement learning (RL) algorithms use a model of the environment to plan and make decisions. The main idea behind model-based RL is that it is more sample-efficient than model-free RL, because it can use the model to plan and simulate future outcomes, rather than relying on trial-and-error to learn the value of actions. Some examples of model-based RL algorithms include:

1. **Dynamic Programming:** This is a classic algorithm for solving model-based RL problems. It uses the model of the environment to compute the optimal value function or policy.
2. **Monte Carlo Tree Search:** This algorithm uses a search tree to plan future actions and uses Monte Carlo simulation to estimate the expected rewards of different actions.
3. **Planning by Dynamic Programming:** This algorithm uses a model of the environment to plan a sequence of actions that will lead to the highest expected reward.
4. **Guided Policy Search:** This algorithm combines model-based and model-free approaches, by using a model to guide the search for a good policy, and then fine-tuning the policy using trial-and-error.

5. **Model Predictive Control:** This algorithm uses a model of the environment to predict the future outcomes of different actions and selects the action that leads to the best outcome.
6. **Linear Quadratic Regulator:** This is a classical control algorithm that uses a model of the environment to compute the optimal control policy.

Advantages of Model-based RL algorithms:

1. They can be more sample efficient than model-free algorithms because they use a model of the environment to plan and simulate future outcomes.
2. They can be more accurate than model-free algorithms because they use a more accurate representation of the environment.
3. They can be more reliable than model-free algorithms because they do not rely on trial and error to learn the value of actions.
4. They can be more flexible than model-free algorithms because they can handle different types of environments, such as continuous or high-dimensional state spaces.

Disadvantages of Model-based RL algorithms:

1. They may require more computational resources to learn and use a model of the environment.
2. They may be more difficult to implement than model-free algorithms because they often require more complex mathematical and programming techniques.
3. They may perform poorly if the model of the environment is inaccurate or incomplete.
4. They may not work well when the environment is stochastic or non-stationary.

2.4.2.2 MODEL FREE REINFORCEMENT LEARNING ALGORITHMS.

Model-free RL algorithms are a class of RL algorithms that do not require a model of the environment's dynamics. Instead, these algorithms rely on the agent's experience to learn the optimal policy. Model-free algorithms are typically more sample-efficient than model-based algorithms, meaning that they require fewer interactions with the environment to learn an optimal policy.

Advantages of Model-free RL algorithms:

1. They can be simpler to implement than model-based algorithms because they do not require a model of the environment.
2. They can learn from experience, so they do not require explicit knowledge of the environment.
3. They can work well when the environment is stochastic or non-stationary.

Disadvantages of Model-free RL algorithms:

1. They may be less sample efficient than model-based algorithms because they rely on trial-and-error to learn the value of actions.
2. They may be less accurate than model-based algorithms because they use a less accurate representation of the environment.
3. They may be less reliable than model-based algorithms because they rely on trial-and-error to learn the value of actions.
4. They may not work well in high-dimensional or continuous state spaces.

Model-free reinforcement learning (RL) algorithms are a class of RL algorithms that do not require a model of the environment's dynamics. Instead, these algorithms rely on the agent's experience to learn the optimal policy. Model-free algorithms are typically more sample-efficient than model-based algorithms, meaning that they require fewer interactions with the environment to learn an optimal policy.

For a detailed understanding of the RL algorithms one can refer to the excellent book by Szepesvari (Szepesvári, 2010a). For a detailed view into deep reinforcement learning algorithms, refer to Li (Y. Li, 2017) and (Arulkumaran et al., 2017)

The Reinforcement problems can be solved using the following methods.

1) Dynamic programming (DP) method,
2) Monte Carlo (MC) method, and
3) Temporal difference (TD) method

are the most essential reinforcement algorithms that are employed to tackle every MDP problem. Figure 2.5 depicts the topology of RL algorithms.

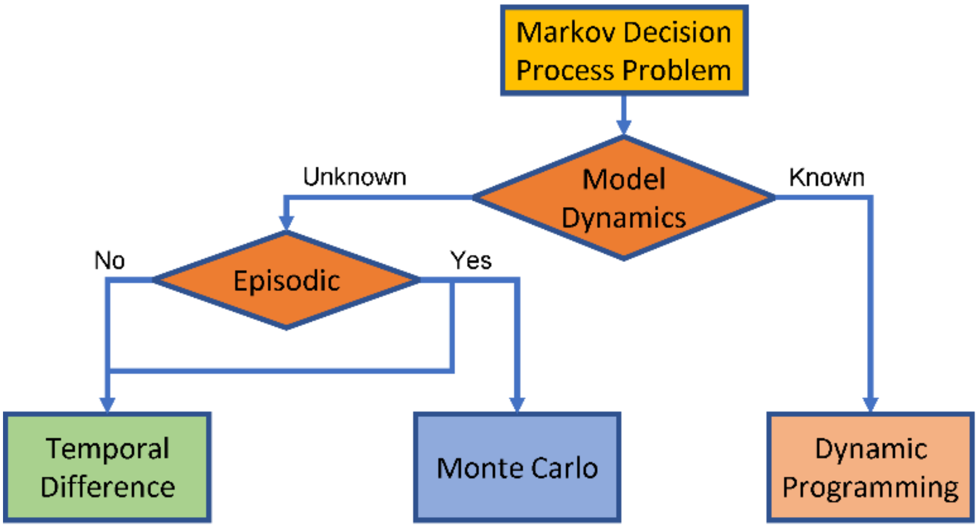


Figure 2.5: Topology of reinforcement learning algorithms.

The DP algorithm is the one that should be used whenever the model dynamics, such as the transition probability and reward probability, are already known. In DP, we are required to have a model of the environment that is both comprehensive and accurate. It is contingent on all the states that came before it. The challenges with the MDP can either be non-episodic or continuous jobs, or they could be episodic. Episodic tasks always reach a conclusion or final state.

The DP can be utilised for activities that are either non-episodic, or continuous, as well as those that are episodic. The only requirement for the MC algorithm is previous experience, in the form of sample sequences of states, actions, and rewards gleaned from either real-world or simulated engagement with an environment. It does not require any prior knowledge of the environment, in contrast to DP. Because TD algorithms do not require a model, the agent is able to learn how to forecast the expected value of a variable that will occur at the conclusion of a series of states.

The TD approach makes it possible to use the acquired state-values as a basis for guiding actions that ultimately alter the state of the environment. Because the case study that will be used to develop the algorithm is not episodic, the MC technique will not be investigated as part of this body of research work. The case study is resolved with both the DP and TD methodologies respectively.

2.4.2.3 Q LEARNING AS MODEL FREE ALGORITHM IN TD

Q-Learning is a popular model-free reinforcement learning (RL) algorithm that uses temporal-differencing to estimate the Q-value, which is the expected long-term reward for taking a specific action in each state. The Q-value is used to determine the optimal policy for a Markov Decision Process (MDP).

In Q-Learning, the Q-value is estimated using the temporal-difference error, which is the difference between the estimated value of a state-action pair and the actual value of that pair, as determined by the rewards received during subsequent time steps. The Q-value estimate for a state-action pair is updated based on this temporal-difference error.

2.4.2.4 BELLMAN EQUATION FOR OPTIMALITY

The Bellman equation is a fundamental concept in the theory of Markov Decision Processes (MDPs) and is used to describe the relationship between the value of a state and the value of the states that can be reached from it. MDPs are a type of mathematical model used to describe decision-making problems where the outcome is uncertain and the state of the system changes over time.

In an MDP, a decision-maker (or agent) interacts with an environment by choosing actions to take at each time step. The environment responds by transitioning to a new state and providing a reward. The agent's goal is to choose actions that will maximize the expected total reward over time.

Methods to solve bellman equation.

There are several methods to solve the Bellman equation. Some of the most common methods include:

1. Dynamic Programming: Bellman equation can be solved using dynamic programming methods such as value iteration or policy iteration. These methods involve iteratively updating the value of each state until convergence is achieved.
2. Monte Carlo Methods: Bellman equation can also be solved using Monte Carlo methods such as first-visit Monte Carlo or every-visit Monte Carlo.

These methods involve averaging the returns over multiple episodes of the game.

3. Temporal Difference Learning: Bellman equation can be solved using temporal difference (TD) learning methods such as Q-learning or SARSA. These methods involve updating the value of a state based on the difference between the current value and the predicted value.
4. Neural Networks: Bellman equation can also be solved using neural network-based methods such as deep Q-networks (DQNs) or actor-critic models. These methods involve using neural networks to approximate the value function or the policy function.
5. Hybrid Methods: A combination of the above methods can also be used to solve the Bellman equation in reinforcement learning. This can be useful when the environment is complex or when the data is limited.

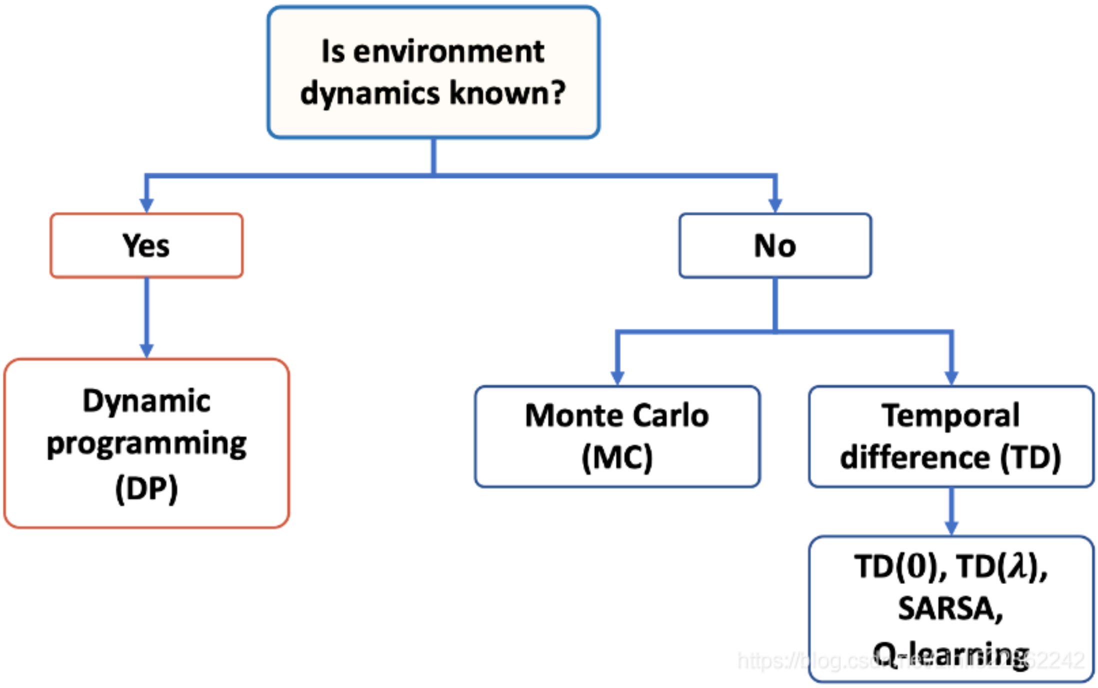


Figure 2.6: Flow chart describing the algorithms (Lin, 2021)

Figure 2.6 describes the use of appropriate RL algorithm. Dynamic Programming, Monte Carlo and Temporal Difference are all methods for solving the Bellman equation in reinforcement learning, but they differ in their approach and assumptions.

2.4.2.5 DYNAMIC PROGRAMMING

Dynamic programming methods such as value iteration and policy iteration involve iteratively updating the value of each state until convergence is achieved. These methods rely on the ability to model the environment, which means that the agent needs to have a complete understanding of the environment's dynamics. Dynamic programming methods are useful for problems with a deterministic environment and/or small state space.

2.4.2.6 TEMPORAL DIFFERENCE

Temporal difference methods such as Q-learning and SARSA involve updating the value of a state based on the difference between the current value and the predicted value. These methods do not require a model of the environment, but they do require the agent to be able to experience multiple episodes of the game. Temporal difference methods are useful in both deterministic and stochastic environments.

In summary, Dynamic Programming methods are useful for problems with a deterministic environment and/or small state space, Monte Carlo methods are typically used in fully observable environments where the agent can experience multiple episodes of the game, and Temporal Difference methods are useful in both deterministic and stochastic environments.

2.4.3 RELATED LITERATURE FOR REINFORCEMENT LEARNING

Machine learning, which falls under the umbrella of artificial intelligence, is the centre of several promising algorithms developed in recent years. These algorithms have helped to restructure the approach that is taken to solving problems. Reinforcement learning, often known as RL, is a methodology that engages in interaction with its surroundings, in contrast to supervised learning and unsupervised learning methods. RL tries to simulate the problems in a way that is like how humans go about their daily lives (Sutton & Barto, 2018). Learning can be categorised as value-based, policy-based, model-based, or imitation learning, depending on the approach that was taken to acquire knowledge. The methods each have their own distinct versions and gradual advances, and they are driven by a significant collection of obstacles. The subject of reinforcement learning faces a variety of common issues (Barto et al., 2017;

Szepesvári, 2010b), including but not limited to the following: planning, exploration, generalisation, data efficiency, temporal abstractions, training stability, scalability, lifelong learning, credit assignment, and sparse incentives.

The RL algorithms can be broadly broken down into model-based and model-free categories, with the division being determined by the policy search. Model-based approaches are those that employ a model in the process of finding a solution to a problem, whereas model-free methods are those that do not employ any models (Deisenroth, 2011). A few examples of common RL algorithms are dynamic programming, temporal difference learning, Q-learning, Monte Carlo learning, and actor-critical learning. While DP is a model-based approach that already has the MDP and only needs to figure out what to do, TP is a model-free technique that does not require the knowledge of a model of the world. DP is a model-based approach that already has the MDP and only needs to figure out what to do. TP is a simulation-based platform that utilises bootstrapping for its learning (Fenjiro & Benbrahim, 2018). Deep reinforcement learning is a subsection of the study of reinforcement learning that was created because of the emergence of deep learning, which dramatically expedited research in RL utilising deep learning algorithms (Arulkumaran et al., 2017; Paraschos et al., 2020). RL algorithms that get their motivation from behavioural psychology learn from trial and error by interacting with the environment's random variables.

The concept of the Markov decision process is the foundation upon which RL issues are constructed (MDP). The presence or lack of some reward is used to inform RL's learning about how best to optimise the agent's behaviour. Over the course of the last ten years, there has been an increase in the amount of light shed on reinforcement learning because of developments in computing power and mathematical techniques. Research on RL has been conducted across a wide variety of subfields, including game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, and statistics.

In the RL-based strategy that they proposed, Paraschos and his team (Paraschos et al., 2020) established the best possible policy for optimal joint control. Estimating the quality inspection process using supervised machine learning models is common practise in the manufacturing industry. In this research, an attempt is made to use reinforcement learning to optimise a quality prediction model that was constructed based on the RFID data that was gathered in a manufacturing pipeline for automobiles. In any manufacturing plant, the process of entering inspection has the goals of controlling quality, lowering production costs, getting rid of scrap, and reducing or eliminating process failure downtimes caused by non-conforming raw materials. The prediction of the raw material acceptance rate can help to govern the selection of raw material suppliers and enhance the manufacturing process by removing non-conformities.

Two different reinforcement learning strategies are utilised to optimise the MDP model for part quality inspection.

2.4.4 RESEARCH GAP IN REINFORCEMENT LEARNING

The authors from the literature have thrown light on various areas of manufacturing to solve the issues pertaining to quality. There are a few areas where research still needs to focus on. Table 2.2 provides the knowledge and the research gap from the literature. Reinforcement learning (RL) using Markov Decision Processes (MDP) is a promising approach for optimizing the quality of vehicles and components in Automotive Quality 4.0. However, there are several research gaps that need to be addressed to fully realize its potential.

One research gap is the integration of RL with other Quality 4.0 technologies, such as Industry 4.0, Internet of Things (IoT), and Big Data Analytics. While RL can be used to optimize the production process, it can be further enhanced by integrating it with other technologies to provide a more complete and accurate picture of the production process and to optimize it accordingly.

Another research gap is the scalability and robustness of RL in large and complex systems. As the automotive industry becomes more complex and more data-driven, it becomes increasingly important to develop RL algorithms that can handle large and complex systems and that can be robust to changes and disturbances in the production process.

Another gap is the integration of RL with real-time monitoring and control systems. Real-time monitoring and control systems allow companies to track the production process in real-time and make adjustments as needed. However, the integration of RL with real-time monitoring and control systems is still an open problem.

Additionally, research gaps exist in the development of efficient algorithms for solving large-scale RL models, as well as the development of methods for validating the results obtained from RL models.

Another gap is the lack of understanding of how RL can be used to optimize Quality 4.0 metrics such as reliability, maintainability, and safety. These metrics are important for ensuring the quality of vehicles and components, but research is needed to understand how RL can be used to optimize them.

Eventually, there is a significant potential for RL using MDP in Automotive Quality 4.0, but research is needed to fully realize this potential by addressing the research gaps such as the integration with other Quality 4.0 technologies, scalability and robustness, real-time monitoring and control systems, efficient algorithms and validation methods, and understanding of how to optimize Quality 4.0 metrics.

2.5 AUTOMATED QUALITY INSPECTION

Automated quality inspection in the automotive industry refers to the use of technology such as robots, machine vision systems, and sensors to perform quality checks on vehicles and automotive components. This can include checks for defects, compliance with standards, and proper assembly. Automated inspection can increase efficiency, accuracy, and consistency compared to manual inspection methods. It can also help to reduce costs and improve safety in the manufacturing process.

Automated visual quality inspection in the automotive industry refers to the use of machine vision systems to perform quality checks on vehicles and automotive components. These systems use cameras and image processing algorithms to analyse the appearance of the parts and detect any defects or deviations from the expected standard. This can include checks for scratches, dents, misalignment, missing parts, and other issues that can affect the performance or aesthetics of the vehicle. Automated visual inspection can be faster and more accurate than manual inspection methods, and it can also be used in areas that are difficult or dangerous for human operators to access. The use of this technology can help to improve the quality of the final product and reduce the cost of the manufacturing process.

Drury and Watson (Drury & Watson, 2002) highlighted the human factors involved the good practices in visual inspection. See (See, 2012) presented a detailed literature review on the visual inspection. He further investigated on the Visual inspection reliability for precision manufacturing parts (See, 2015).

Johson and his associates (T. L. Johnson et al., 2019) detailed the application of visual inspection in manufacturing, while Huang and Pan (S. H. Huang & Pan, 2015) surveyed the automated visual inspection in semiconductor industry.

2.5.1 IMAGE CLASSIFICATION USING DEEP (PASSIVE) LEARNING

Automotive visual quality inspection using machine learning involves using image recognition and computer vision techniques to automatically inspect the quality of vehicles or components. This can include identifying defects, such as scratches, dents, or misaligned parts, as well as ensuring that the finished product meets certain standards, such as colour and shape.

One common approach to automotive visual quality inspection using machine learning is to use supervised learning to train a model to recognize defects in images of vehicles or components. This typically involves providing the model with a dataset of images that have been labelled with the presence or absence of defects and using this dataset to train the model to recognize defects in new images. Once the model is trained, it can be used to automatically inspect new images and flag any defects that are detected.

Another approach is using unsupervised learning, where a model is trained on an unlabelled dataset, meaning that the correct output is not known for each input. This type of machine learning can be used to identify patterns and anomalies in the production process, such as identifying which machines or operators are causing the most defects.

Deep learning techniques such as convolutional neural networks (CNNs) are also commonly used for visual quality inspection in the automotive industry. CNNs are particularly well-suited for image recognition tasks and can be trained to detect and classify defects in images of vehicles or components with high accuracy.

Overall, machine learning techniques can be used to automate the visual quality inspection process in the automotive industry, reducing the need for manual inspection and increasing the speed and accuracy of the inspection process.

2.5.2 NEED FOR ACTIVE LEARNING

During any traditional deep learning-based model training, the first step is to prepare a limited number of supervised data that are randomly chosen from a pool of unlabelled data and are then labelled by a person. These data are then used to train the model. The process is named as batch learning or formally “passive learning” (PL)(Jo et al., 2022; Szepesvári, 2010b) .

Deep learning has been so successful because of the massive amounts of data that have been supervised. There have been reports in recent research that increasing the quantity of data available to a model can increase its overall performance (Sun et al., 2017). However, obtaining a massive amount of supervised data in real-world applications is problematic because to the constraints of time and money (Jo et al., 2022).

In the relatively new discipline of deep learning, active learning has just emerged as one technique to acquire supervised data in an effective manner. Instead of selecting the data to be labelled at random, active learning uses an algorithm to actively select the data to be labelled, with the overarching goal of efficiently improving the performance of the target model. This contrasts with traditional learning, which uses random selection of the data to be labelled (Jo et al., 2022).

Active learning approach tries to circumvent the labelling bottleneck by posing questions in the form of unlabelled examples that are then sent to an oracle (such as a human annotator) for classification. In this manner, the goal of the active learner is to acquire a high level of accuracy while utilising a minimum number of labelled examples to cut down on the expense of getting labelled data.

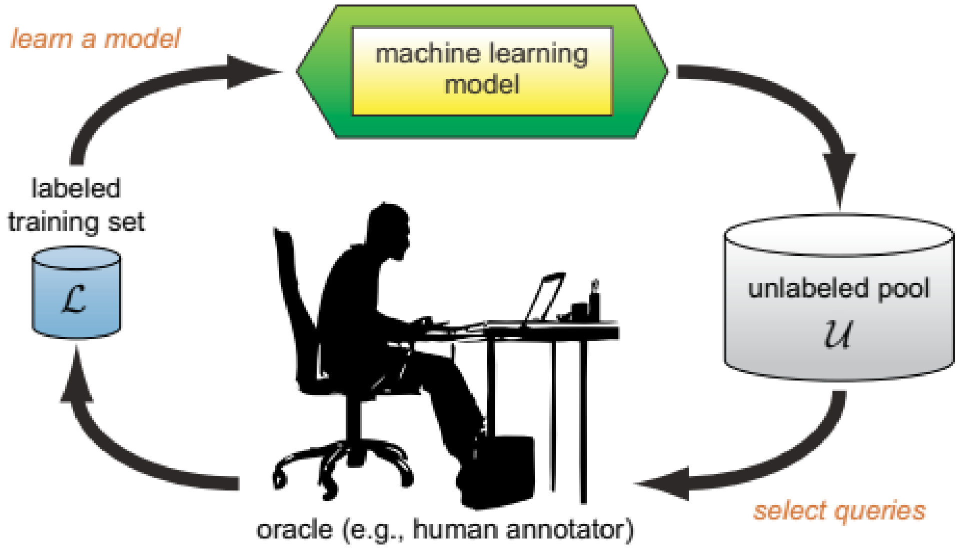


Figure 2.7: A pool based active learning cycle (Settles, 2009)

Figure 2.7 depicts a typical pool based active learning cycle. Settles (Settles, 2009) conducted a detailed literature survey on Active Learning.

2.5.2.1 DISADVANTAGES OF PASSIVE LEARNING

Passive learning, also known as batch learning, has several disadvantages compared to active learning. Some of the main disadvantages are:

1. **Lack of adaptability**: Passive learning algorithms are not able to adapt to changes in the data distribution, as they only use a fixed set of labelled data to train the model.
2. **High annotation cost**: Passive learning can be expensive when the cost of annotation is high, as all instances need to be labelled.
3. **Data imbalance**: Passive learning can lead to poor performance on under-represented classes, as the model only learns from the fixed set of labelled data.
4. **Overfitting**: Passive learning algorithms can easily be overfit to the training data, as they do not have a mechanism for selecting informative instances to be labelled.
5. **Limited performance**: Passive learning algorithms may not reach the same level of performance as active learning algorithms, as they do not actively select instances to be labelled.
6. **Lack of Exploration**: Passive learning algorithms do not explore the feature space and may miss important patterns in the data.

In detail, Passive learning has several disadvantages such as lack of adaptability to changing data, high annotation cost, data imbalance, overfitting, limited performance, and lack of exploration. Passive learning is often used when labelled data is abundant and easily available or when the cost of annotation is low.

2.5.2.2 ACTIVE LEARNING VS PASSIVE LEARNING

Active learning, also known as Human in the loop learning, has significant advantages compared to passive learning.

1. Active learning involves actively selecting the instances to be labelled in order to improve the performance of the model. In active learning, the algorithm selects instances that are most informative and likely to improve the classifier's performance, and then queries an oracle (such as a human expert) to label those instances. ***The goal of active learning is to learn a classifier using the least amount of labelled data while ensuring that the classifier has high accuracy***.
2. Passive learning, on the other hand, involves using a fixed set of labelled data to train a model. In passive learning, the algorithm does not actively select the instances to be labelled, but instead uses all the labelled data that is available. ***The goal of passive learning is to learn a classifier from the given labelled data set***.
3. Active learning is often used when labelled data is scarce or expensive, the cost of annotation is high, data imbalance is present, to improve the performance of the model and when human expertise is needed to label the data. ***Passive learning, on the other hand, is often used when labelled data is abundant and easily available or when the cost of annotation is low***.

To describe, Active learning involves actively selecting instances to be labelled to improve the performance of the model, while passive learning involves using a fixed set of labelled data to train a model, without actively selecting instances to be labelled.

2.5.3 NEED FOR DEEP LEARNING AND ACTIVE LEARNING

While DL has a great learning capability in the context of high-dimensional data processing and automatic feature extraction, AL offers a substantial potential to successfully minimise labelling costs. DL has a strong learning power in the situation of automatic feature extraction. Consequently, one apparent solution is to merge DL and AL, as doing so will significantly broaden the applications to which they might be put.

DeepAL is the name given to this combined technique, which was created after taking into consideration the advantages that the two methodologies provide to one another. Researchers have great hopes for the outcomes of studies that are being conducted in this area. However, despite the abundance of research that has been done on query approach in relation to AL, it is still extremely challenging to implement this method directly into DL.

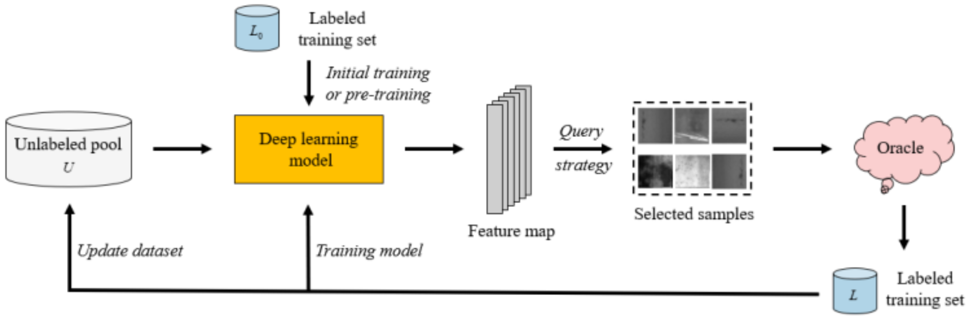


Figure 2.8: A typical example of deep active learning (Ren et al., 2022)

Figure 2.8 shows a typical deep active learning approach. In a different focus from models to data, Sun, Shrivastava, Singh, and Gupta (Sun et al., 2017) highlighted that relationship between the data and visual deep learning. Their studies focussed on various parameters of deep learning such as the use of large volumes of data, model capacity on various computer vision tasks.

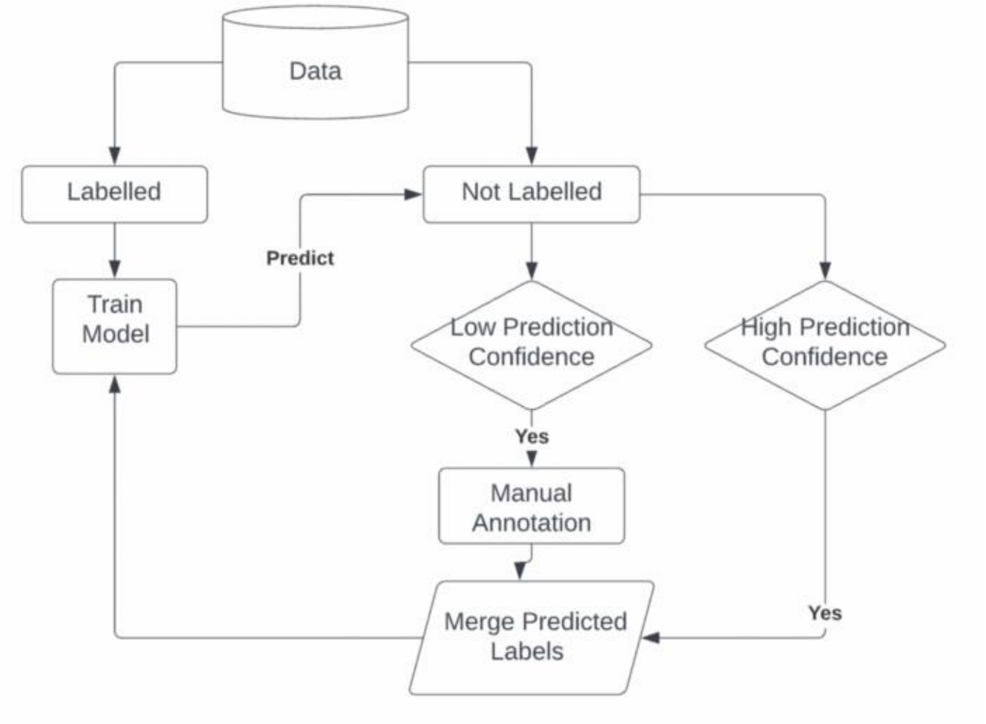


Figure 2.9: Active learning framework (Vatsal, 2022)

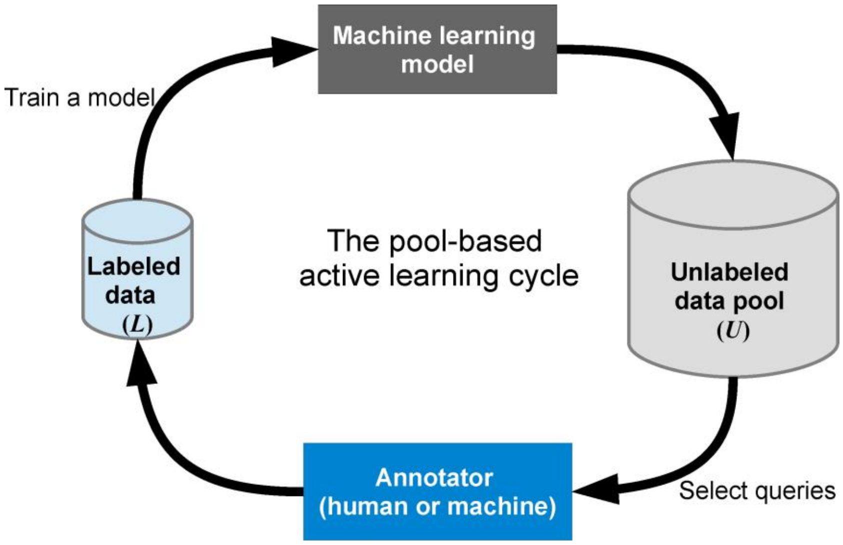

Machine learning model
Train a model
The pool-based active learning cycle
Labeled data (*L*)
Unlabeled data pool (*U*)
Annotator (human or machine)
Select queries

Figure 2.10: A typical example of deep active learning

2.5.4 QUERY STRATEGIES

This section demonstrates the deep active learning techniques available for the image classification. The following are few of the sampling algorithms available in the literature.

1. Entropy algorithm

2. Random algorithm
3. Uncertainty algorithm
4. Deep Bayesian Active Learning (DBAL) algorithm
5. Bayesian Active Learning by Disagreement (BALD) algorithm

2.5.4.1 ENTROPY SAMPLING

Entropy sampling is a query strategy that can be used in visual inspection tasks to select the most informative images to inspect. The goal of entropy sampling is to select images that have the highest entropy, which is a measure of the uncertainty or randomness of the image.

Entropy is used as a measure of uncertainty because it is a measure of the amount of information needed to specify the state of a system. In the case of visual inspection, the state of the system is the presence or absence of a defect in an image. An image with high entropy has a high degree of uncertainty as to whether or not it contains a defect, while an image with low entropy is more likely to contain a defect or not.

Entropy sampling can be used in a variety of visual inspection tasks, such as quality control in manufacturing, medical imaging, and object detection in autonomous vehicles.

The basic idea of entropy sampling is to select images that are most uncertain or least certain, and then inspect those images to reduce the uncertainty about the presence or absence of a defect.

Entropy sampling has been shown to be effective in reducing the number of images that need to be inspected while still maintaining a high level of accuracy in detecting defects. This can help to reduce the cost and time required for visual inspection tasks, and improve the efficiency of the inspection process.

It's important to note that entropy sampling is not a panacea and the effectiveness of the method depends on the specific problem and the data distribution. Additionally, the computation cost of entropy sampling can be high, which may limit its use in real-time applications.

2.5.4.2 RANDOM SAMPLING

Random sampling is a query strategy that can be used in visual inspection tasks to select images for inspection. The basic idea behind random sampling is to select images for inspection at random, rather than using any specific criterion to choose which images to inspect.

Random sampling can be useful in visual inspection tasks because it is a simple and unbiased method for selecting images for inspection. It does not require any prior knowledge about the distribution of defects in the images, and it can be used with a wide variety of image types and inspection tasks.

One of the main advantages of random sampling is that it can be used in situations where the distribution of defects is unknown or difficult to estimate. It can also be used as a benchmark method against which other query strategies can be compared.

However, one of the main disadvantage of random sampling is that it can be inefficient, and it may not always be the best strategy for selecting images for inspection. Since it does not take into account any information about the images or the presence of defects, it may lead to a higher number of images to inspect in order to achieve a certain level of accuracy.

In conclusion, random sampling is a simple and unbiased query strategy that can be useful in visual inspection tasks, but it may not be the most efficient method, especially when the distribution of defects is unknown or difficult to estimate. Other query strategies such as entropy sampling, active learning, or Bayesian optimization may be more efficient and better suited for specific problems.

2.5.4.3 UNCERTAINTY SAMPLING

Uncertainty sampling is a query strategy that can be used in visual inspection tasks to select images for inspection. The basic idea behind uncertainty sampling is to select images for inspection that have the highest degree of uncertainty or ambiguity, based on the current state of the model.

In visual inspection tasks, uncertainty sampling can be used to identify images that are most likely to contain defects, by selecting images that the model is least certain about. The model's uncertainty can be determined by calculating the entropy or a similar measure of the model's confidence for each image, and selecting images with the highest uncertainty.

Uncertainty sampling can be useful in visual inspection tasks because it can help to improve the performance of the model by selecting images that are most informative for training. By focusing on images that the model is least certain about, uncertainty sampling can help to reduce the number of images that need to be inspected while still maintaining a high level of accuracy in detecting defects.

One of the main advantages of uncertainty sampling is that it can be used in conjunction with other query strategies and active learning algorithms, and can be easily integrated into existing machine learning models.

However, one of the main disadvantage of uncertainty sampling is that it can be computationally expensive and may not be suitable for real-time applications. Additionally, the effectiveness of the method depends on the specific problem and the data distribution.

In conclusion, uncertainty sampling is a query strategy that can be useful in visual inspection tasks, by selecting images that the model is least certain about, it can help to improve the performance of the model by selecting images that are most informative for training. However, it can be computationally expensive and may not be suitable for real-time applications, and its effectiveness depends on the specific problem and data distribution.

2.5.4.4 BALD SAMPLING

BALD (Bayesian Active Learning by Disagreement) sampling is a query strategy that can be used in visual inspection tasks to select images for inspection. The basic idea behind BALD sampling is to select images for inspection that have the highest expected reduction in uncertainty, based on the current state of the model.

In visual inspection tasks, BALD sampling can be used to identify images that are most informative for training the model, by selecting images that have the highest expected reduction in uncertainty. BALD sampling takes into account both the model's uncertainty and the diversity of the selected images, this way it balances the exploration-exploitation trade-off, and it can help to reduce the number of images that need to be inspected while still maintaining a high level of accuracy in detecting defects.

BALD sampling can be useful in visual inspection tasks because it can help to improve the performance of the model by selecting images that are most informative for training. It's also able to balance the exploration-exploitation trade-off, by considering both the model's uncertainty and the diversity of the selected images.

One of the main advantages of BALD sampling is that it can be used in conjunction with other query strategies and active learning algorithms, and it can be easily integrated into existing machine learning models.

However, one of the main disadvantage of BALD sampling is that it can be computationally expensive and may not be suitable for real-time applications. Additionally, the effectiveness of the method depends on the specific problem and the data distribution.

In conclusion, BALD sampling is a query strategy that can be useful in visual inspection tasks, by selecting images that have the highest expected reduction in uncertainty, it can help to improve the performance of the model by selecting images that are most informative for training, and it also balances the exploration-exploitation trade-off. However, it can be computationally expensive and may not be suitable for real-time applications, and its effectiveness depends on the specific problem and data distribution.

2.6 RESEARCH GAP

Authors in the literature have presented various studies and approaches to determine the quality management. Industries spend 30-40% of the total production cost for quality inspection.

Active learning is a promising approach for automating the visual quality inspection process in Automotive Quality 4.0. However, there are several research gaps that need to be addressed to fully realize its potential.

The implementation of Quality 4.0 using deep active learning in the automotive industry is a relatively new and active area of research. While many studies have shown the potential of deep active learning in improving the quality of vehicles and components, there are still several research gaps that need to be addressed in order to fully realize its potential in the Automotive Quality 4.0.

One void in the research is the integration of deep active learning with other Quality 4.0 technologies such as Industry 4.0, Internet of Things (IoT), and Big Data Analytics. While deep active learning can be used to improve the performance of visual quality inspection models, it can be further enhanced by integrating it with other technologies to provide a more complete and accurate picture of the production process and to optimize it accordingly.

Another research gap is the scalability and robustness of deep active learning in large and complex systems. As the automotive industry becomes more complex and more data-driven, it becomes increasingly important to develop deep active learning algorithms that can handle large and complex systems, and that can be robust to changes and disturbances in the production process.

Additionally, research gaps exist in the development of efficient algorithms for solving large-scale deep active learning models, as well as the development of methods for validating the results obtained from deep active learning models.

Another gap is the lack of understanding of how deep active learning can be used to optimize Quality 4.0 metrics such as reliability, maintainability, and safety. These metrics are important for ensuring the quality of vehicles and components.

Research gaps in implementation of Active learning in automotive Quality 4.0

Forecast modelling and automation of quality inspection can provide reduced cost and production life cycle which is highlighted in the literature. The summary of the research findings in literature is summarized in Appendix A.

2.7 CHAPTER SUMMARY

In Chapter 2, the theory and literature related to this study's key themes are broadly addressed. Reinforcement learning, deep active learning, quality inspection, and the Markov chain are important subjects. Relevant sections of the chapter are separated. Quality management, including supply chain and entry inspection quality, is covered in Section 2.2. This part will also include supply chain quality management's latest research. Next, the traditional supply chain's issues are shown. Markov chains and the production process are discussed in section 2.3. Detailing Markov chain characteristics. Then, existing research is reviewed, and its findings emphasised. The Markov decision process and reinforcement learning's core theories and techniques are presented in section 2.4. The Markov decision process can be solved via reinforcement learning. Recent research is critically assessed, and knowledge gaps are identified. In Section 2.5, the principles of automated quality inspection, active learning requirements, and deep active learning approach are explained. The study's and literature's shortcomings are detailed in Section 2.6. The following chapter presents the research methods employed to address the objectives of this research work.

CHAPTER 3

CHAPTER THREE: RESEARCH METHODOLOGY

3.1 CHAPTER INTRODUCTION

The study approach is broken down into three distinct sections and discussed in depth in Chapter 3. In the next section (3.2), the Markov decision procedure and the inbound inspection case study utilising RFID tag data are presented to the reader. In order to provide clarity on the case study, a state machine diagram has been constructed. The case study for the inbound inspection and the substations that were involved are both explained in full. In this part, the computation of Markov chain characteristics for the experimental cases is demonstrated in additional detail. Additionally, the transition matrix as well as an assessment of the transition probability as well as the processes that were utilised for the creation of the raw material acceptance prediction model are described in this part. The prediction model that was constructed using the Markov decision process that was solved in the previous sections is optimised in Part 3.3 through the use of reinforcement learning, which is explained in detail in that section. This section then goes on to discuss the actions that need to be taken in order to solve the raw material acceptance prediction model using the temporal difference RL algorithm and the dynamic programming technique. In a similar fashion, this section details the process that must be followed in order to create a model for the estimate of the material quality classification utilising the temporal difference technique. The impeller flaws casting dataset is described in full in section 3.4, along with the standard deep learning technique to visual inspection. This section goes into additional depth about the procedures that need to be taken in order to maximise the model development by making best use of the data based on a deep active learning method. Lastly, the approaches discussed in this chapter are summed up in section 3.5, which also provides a link to the chapter on the outcomes.

3.2 MARKOV DECISION PROCESS (MDP)

A Markov Decision Process (MDP) is a mathematical framework for modelling decision-making situations in which the outcomes are uncertain. It is an extension of a Markov chain, in which the system can be in different states and transitions between states are governed by certain probabilities. The goal of an MDP is to find the optimal policy, which is a function that maps states to actions and maximizes the expected cumulative reward over time. This can be done using dynamic programming techniques such as value iteration or policy iteration.

MDPs are widely used in a variety of fields, including operations research, artificial intelligence, and control systems. In the automotive industry, MDPs can be used to model and optimize decision-making in areas such as production planning, inventory management, and logistics.

Additionally, MDPs can also be used for modelling the failure of the systems and find the optimal maintenance policy to increase the reliability of the systems. It could also be used for modelling the decision-making process of the autonomous vehicles, where the decision maker is the vehicle itself.

3.2.1. RMAP MODEL DEVELOPMENT

Based on the RFID data that is collected throughout the manufacturing process of an automobile, a model for quality prediction is constructed. An RFID reader is often mounted in each substation in a factory that manufactures automobiles. This reader keeps a record of the various components as they go through the substations. The information gleaned from the RFID reader is utilised in its raw form. The weights for the Markov analysis are estimated every day based on the daily performance, and the details of this process can be found in section 4.2

3.2.2. INCOMING INSPECTION IN AUTOMOTIVE SUPPLY CHAIN

The process flow diagram of an automotive supply chain environment's entering inspection station is depicted in figure 3.2. In contexts related to the automotive supply chain, this diagram is displayed. The workstation is partitioned into seven distinct zones: the packaging inspection, the visual inspection, the gauge inspection, the rework station 1, the rework station 2, the return, and the pack and store portions. Each of these zones is responsible for a specific type of inspection. It is anticipated that the material that is acquired from the supplier will satisfy the material specifications that have been established. A series of visual and dimensional inspections are carried out on the material in question in order to ascertain whether or not it satisfies the requirements necessary to qualify.

The individual functions of each of the seven substations, as well as the overall operation of the substations, are depicted in Figure 3.2. In order to accommodate any minor adjustments that may occur, the rework stations will be responsible for making modifications to the materials. After the adjustments at rework 1 and rework 2 have been made, the materials will then be sent for inspection at the visual inspection and the gauge inspection, respectively. After the insignificant flaws have been fixed, they go through a second inspection at the substations where they were originally found. After it has been determined that the defects cannot be remedied, the materials are returned to the vendor through the return gateway substation.

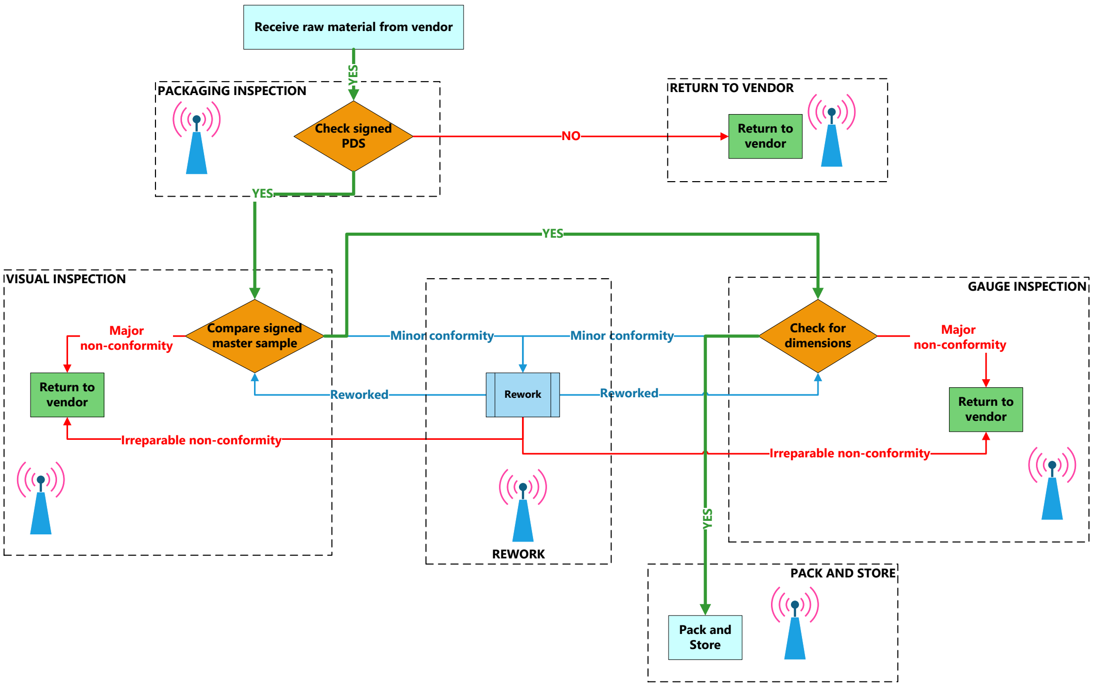


Figure 3.1: Flowchart for the incoming inspection

Raw materials are unloaded from the trucks and placed into storage at the incoming bay. The Packaging and Delivery Standard (PDS) document is signed by both parties, and their respective experts examine the product to determine whether or not it satisfies the predetermined packaging criteria that are outlined in the document. The packages that do not meet the criteria are returned to the vendor, and the inventory is revised so that it accurately reflects the newly acquired knowledge. The packages that have already been accepted are the ones that are subjected to the visual inspection.

During the visual examination, the trained expert removes the item from its packaging and examines it in relation to the standard sample in terms of its colour, texture, and overall appearance. Materials that have only minor flaws are sent back to be reworked as they are considered defective. A second inspection is performed on the materials after they have been revised. They are only allowed to proceed to the next level once they have successfully completed the previous one, which involves having their dimensions evaluated. Following the completion of the rework, the materials that have been altered in an irreversible manner are delivered back to the vendor. In addition, the materials that have significant deformations are returned to the vendor for further inspection. After that, the inventory is revised so that it takes into account both the reworked materials and the items that have been handed back in.

The gauges are then used to determine whether the visually qualified materials fulfil the specified dimension within the allowed tolerance. This determination is made after the materials have been visually qualified. It is essential to address the concerns regarding the fit and finish that have been taking place between mating parts by utilising this method. The components that did not fail the inspection are subjected to a final examination before being stockpiled in the warehouse for later deployment. The major non-conformities are resent to be corrected while the minor non-conformities are being adjusted and rechecked for acceptance. After that, the inventory is then updated to reflect the materials' revised, returned, and OK conditions respectively.

3.2.3 STATE TRANSITION DIAGRAM

Raw materials are unloaded from the trucks and placed into storage at the incoming bay. The Packaging and Delivery Standard (PDS) document is signed by both parties, and their respective experts examine the product to determine whether or not it satisfies the predetermined packaging criteria that are outlined in the document. The packages that do not meet the criteria are returned to the vendor, and the inventory is revised so that it accurately reflects the newly acquired knowledge. The packages that have already been accepted are the ones that are subjected to the visual inspection.

During the visual examination, the trained expert removes the item from its packaging and examines it in relation to the standard sample in terms of its colour, texture, and overall appearance. Materials that have only minor flaws are sent back to be reworked as they are considered defective. A second inspection is performed on the materials after they have been revised. They are only allowed to proceed to the next level once they have successfully completed the previous one, which involves having their dimensions evaluated. Following the completion of the rework, the materials that have been altered in an irreversible manner are delivered back to the vendor. In addition, the materials that have significant deformations are returned to the vendor for further inspection. After that, the inventory is revised so that it takes into account both the reworked materials and the items that have been handed bacin.

Using a finite state machine (FSM) diagram, each of the seven substations that make up the input inspection are disassembled into the component parts that make them up. A representation of a finite state machine that includes each of the seven substations is presented in figure 3.1. Figure 3.1 displays their representation in a machine state diagram for your reference. One way to think of each substation is as a state in the machine.

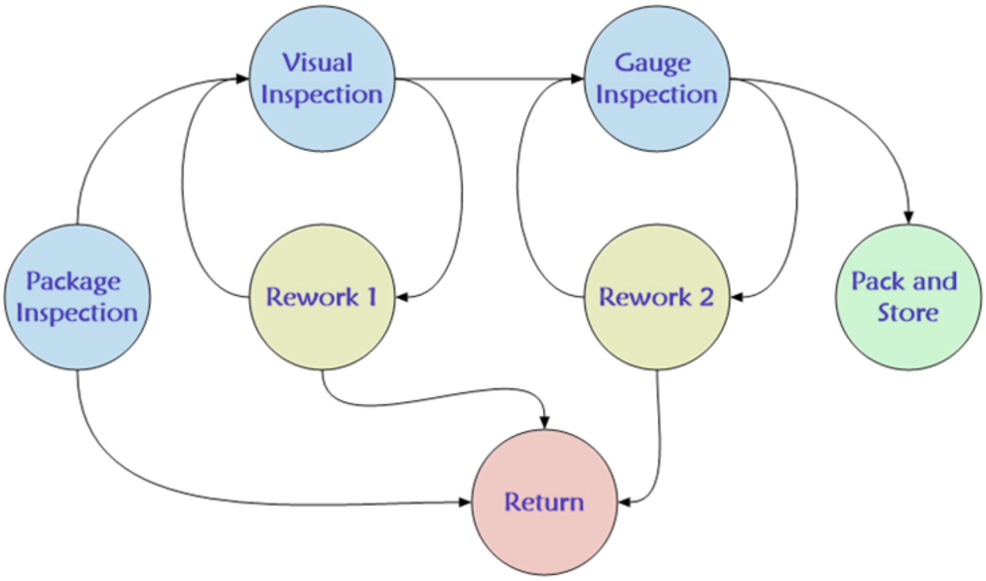


Figure 3.1: FSM Diagram of the states

There are a total of seven distinct states depicted in the transition state diagram that can be found in Figure3.1. To clearly demonstrate the flow of the states from one to the next, arrows have been incorporated into the diagram. The only states that are taken into consideration during the transition to the next state are the current state and the input. Other states are ignored during this phase. It is not taken into account in any way that the state that came before it existed. A tabular representation of the entire transition that occurs between each sub-state is included in Table 3.1, along with a description of that transition. In the column labelled "output," you will find an expression that represents the result of each transition.

Table 3.1. State transition of the incoming inspection of raw material.

	Previous	*Current*	*Input*	*Next*	*Output*	*Description*
PI	-	PI	OK	VI	Conformities	Move to VI
	-	PI	Not OK	RT	Error in packaging	Return to supplier
VI	PI	VI	OK	GI	Conformities	Move to GI
	PI	VI	Not OK	RW1	Minor change	Send to Rework
	RW1	VI	OK	GI	Conformities	Move to GI
	RW1	VI	Not OK	RT	Nonconformities	Return to Supplier
GI	VI	GI	OK	PS	Conformities	Pack and store
	VI	GI	Not OK	RW2	Minor change	Send to Rework
	RW2	GI	OK	PS	Conformities	Pack and Store
	RW2	GI	Not OK	RT	Nonconformities	Return to supplier
RW1	VI	RW1	OK	VI	Minor correction	Move to VI
	VI	RW1	Not OK	RT	Major change	Return to Supplier
RW2	GI	RW2	OK	GI	Minor correction	Move to GI
	GI	RW2	Not OK	RT	Major change	Return to Supplier
PS	GI	PS	OK	-	Conformities	Finish
RT	-	RT	OK	-	Nonconformities	Return to Supplier

3.2.4 MARKOV CHAIN FOR INCOMING INSPECTION

The functions that are carried out in the substations are illustrated through the use of a graphical representation of a finite state machine. In addition to this, once the functions have been translated into it, a Markov chain is constructed from the functions themselves. A discrete-valued implementation of the Markov process is referred to as a "Markov chain" in common parlance. When a Markov chain has discrete values, the state space that contains all the potential values for the chain is either finite or measurable. A Markov process is a type of stochastic process in which the past of the process does not influence the outcome of the process if the current state of the system is known. This type of process is named after the Russian mathematician and statistician Markov. [8-13] (Khoshkangini et al., 2020; Leigh et al., 2017; Lieber et al., 2013; Schmitt et al., 2020) The current state is in possession of all of the information about the past and the present that can be used to make accurate predictions about the future. The only thing that can therefore determine the current state is the one that came before it,

$$\mathrm{p}(\mathrm{S}_\mathrm{n}/\mathrm{S}_{\mathrm{n}-1},\mathrm{S}_{\mathrm{n}-2},\ldots)=\mathrm{p}(\mathrm{S}_\mathrm{n}=\mathrm{s}_\mathrm{n}/\mathrm{S}_{\mathrm{n}-1}=\mathrm{s}_{\mathrm{n}-1});$$

$$\mathrm{s}_\mathrm{n}\in\mathrm{S}=\{\mathrm{s}_1,\mathrm{s}_2,\ldots\mathrm{s}_\mathrm{M}\} \tag{3}$$

The following is an example of a well-known Markov chain, which is also referred to as a first-order Markov chain. In addition to this, if the probabilities of transitions are not reliant on the passage of time n, then,

$$\mathrm{P}\{\mathrm{S}_\mathrm{n}=\mathrm{s}_\mathrm{n}\,|\,\mathrm{S}_{\mathrm{n}-1}=\mathrm{s}_{\mathrm{n}-1}\}=\mathrm{P}\{\mathrm{S}_2=\mathrm{s}_2\,|\,\mathrm{S}_1=\mathrm{s}_1\},\forall\mathrm{s}_\mathrm{n}\in\mathrm{X} \tag{4}$$

3.2.5 STOCHASTIC PROCESS AND MARKOV CHAIN

The study of how the values of a random variable change over the course of time is referred to as the study of a stochastic process. The stochastic process that enables the status of the system to be monitored at discrete points in time is referred to as a "discrete-time stochastic process," which is the meaning of the term "discrete-time stochastic process." During the course of a stochastic process that runs in real-time, it is possible to ascertain the current state of the system at any given instant in time. The transition probability can be represented by an s-by-s matrix.

$$P=\begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1s} \\ p_{21} & p_{22} & \cdots & p_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ p_{s1} & p_{s2} & \cdots & p_{ss} \end{pmatrix} \tag{5}$$

Each row's probability must add up to 1, i.e., for each i

$$\sum_{j=1}^{j=s} p_{ij} = 1$$

(6)

$$P = \left[\begin{array}{cccccccc} & PI & VI & GI & RW1 & RW2 & RT & PS \\ PI & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ VI & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ GI & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ RW1 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ RW2 & 0 & 0 & 1/2 & 0 & 0 & 1/2 & 0 \\ RT & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ PS & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

(7)

3.2.6 PROPERTIES OF MARKOV CHAINS

A Markov chain is a mathematical system that undergoes transitions from one state to another according to certain probabilistic rules. The defining characteristic of a Markov chain is that no matter how the system arrived at its current state, the possible future states are fixed. In other words, the probability of transitioning to any state is dependent solely on the current state and time elapsed.

Here are some properties of Markov chains:

1. Finite state space: A Markov chain must have a finite number of states. In this case study, there are Seven finite states.

2. Memoryless property: The probability of transitioning to any state is dependent only on the current state and time elapsed. The past states do not affect the future states. The probability that a raw material clears gauge inspection (GI) and reaches package and store (PS) totally depends on its performance in visual inspection (VI) and not on packing Inspection (PI).

3. Irreducibility: It is possible to get to any state from any other state in a finite number of steps. In this case study, not all states are reachable form all other states. For instance, gauge inspection (GI) and pack and store (PS) are not reachable form PI. This MDP is Reducible.

4. Periodicity: Some states may be periodic, meaning that the chain will return to the same state after a certain number of steps. Consider visual inspection (VI) and gauge inspection (GI). If an item has minor irregularities, it is sent to rework station for correction. After correction, its again sent to the respective station for error proofing. So, rework station (RW) is a Recurrent state. Visual inspection (VI) and gauge inspection (GI) are communicating states.

5. Aperiodicity: A chain is aperiodic if it is not periodic. This means that it is possible to return to a state, but it will not necessarily happen after a certain number of steps. Pack and store (PS) and return (RT) are aperiodic.

6. Ergodicity: A Markov chain is ergodic if, given enough time, it will visit all states with probability 1. This means that the chain is irreducible and aperiodic. From above discussions, the chain is reducible and aperiodic. So, the chain doesn't exhibit ergodicity.

3.2.7 INITIAL PROBABILITY DISTRIBUTION

After demonstrating an acceptable level of quality at the supplier plant, the raw material is then introduced into the production facility. At the inspection bay, each and every material is examined to determine whether or not it satisfies the PDS standards for its packaging. If it does, the material is allowed to proceed to the next stage of the process. You are welcome to examine Figure 2 to get a better understanding of the state flow diagram as well as the transition probabilities. Before moving on to the subsequent phase of quality control, it must first be determined whether or not the material complies with the PDS. Only then can it be considered for advancement. As a result, the packaging inspection (PI) is the one and only way to gain access to the raw material, and the initial distribution probability P0 can be determined by applying the following formula:

$$P0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

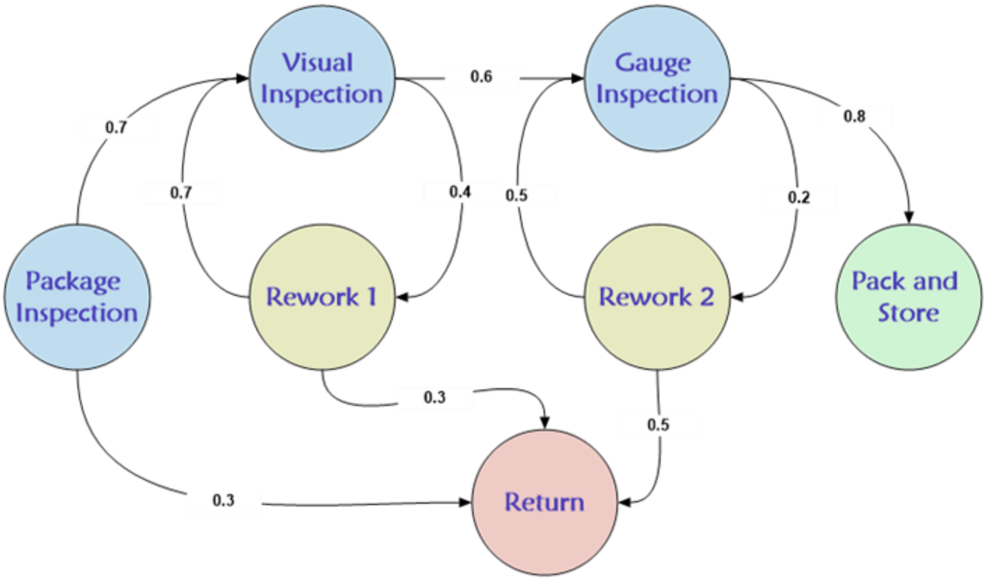


Figure 3.2. State flow diagram with transition probabilities.

3.2.8 TRANSITION MATRIX

In the fields of engineering, operations research, and time series analysis, Markov models are utilised frequently in research and analysis. In this particular piece of writing, it is used to describe transitions between states that occur at the arrival inspection station of a supply chain for automotive assembly.

A matrix in which the number of states is indicated by the letter 'n' is referred to as a transition matrix (abbreviated as P). The matrix is constructed with the help of the transition probability offered by the Markov process. The probability of moving from state 'j' to state I is proportional to each element in the transition matrix denoted by the symbol 'Pij,' which is shown in the form of a 'n x n' matrix. Therefore 0≤P ij≤1 must be true for all ' I ' and ' j '. The probabilities of making the transition between the stations are laid out in Figure 3.2.For study, consider three test cases , A,B and C.The transition probabilities are as explained in Table 3.2 and their state diagrams are shown in Figure A,B and C respectively

Table 3.2. Test cases A, B and C for raw material acceptance and rejection in each state

State - Action	***Case-A (30/70)***	***Case-B (70/30)***	***Case-C (50/50)***
PI-VI	0.3	0.7	0.5
PI-RT	0.7	0.3	0.5
VI-GI	0.4	0.6	0.5
VI-RWI	0.6	0.4	0.5
GI-RW2	0.8	0.2	0.5
GI-PS	0.2	0.8	0.5
RWI-VI	0.4	0.6	0.5
RW1-RT	0.6	0.4	0.5
RW2-GI	0.2	0.8	0.5
RW2-RT	0.8	0.2	0.5

For the test cases, the state diagrams are shown below.

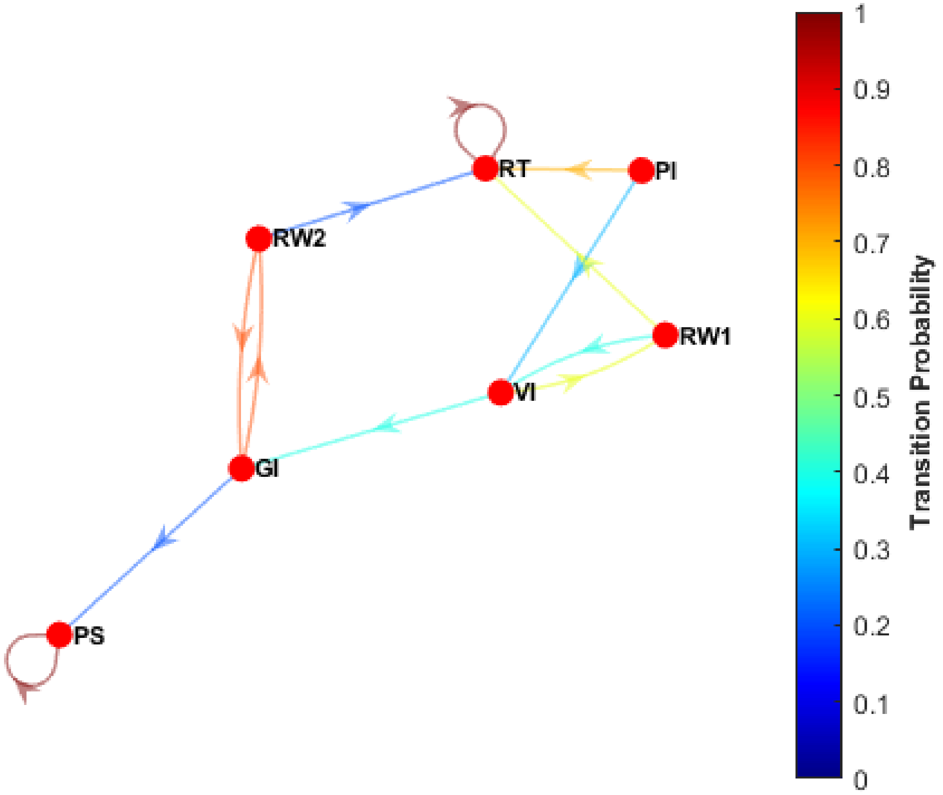


Figure 3.3.a: State diagram for case A

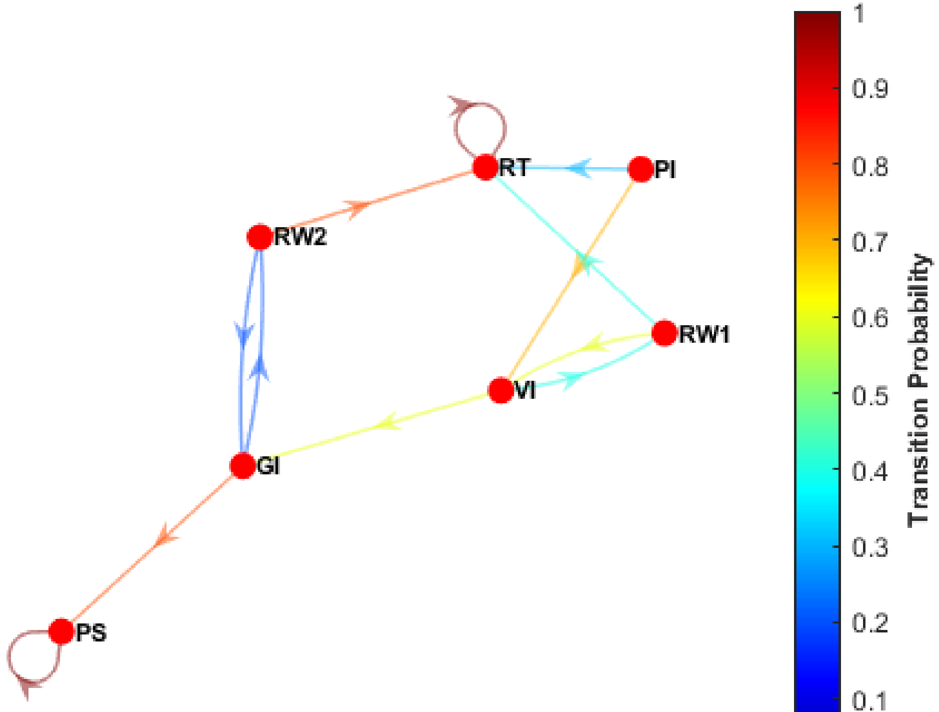


Figure 3.3.b : State diagram for case B

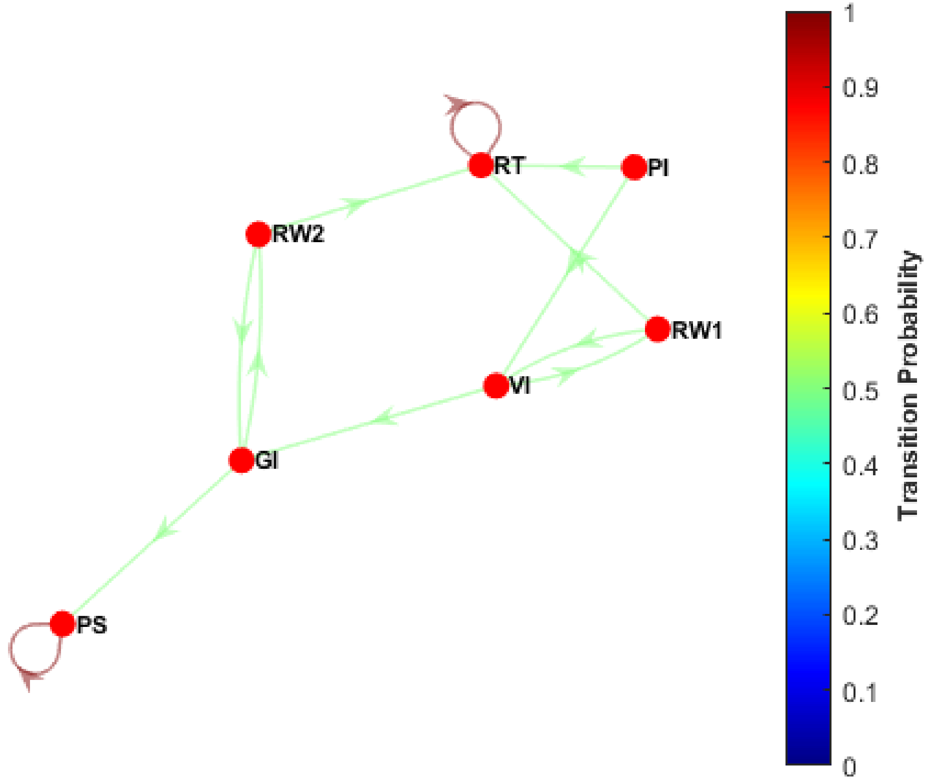


Figure. 3.3.c: State diagram for case C

The developed model is computed to estimate the following results.

a. Probability of the raw material being accepted or rejected.

b. Forecasting the probability of the raw material at an arbitrary state

c. Estimate the steady-state probability for the given transition matrix.

3.2.9 N -STEP TRANSITION PROBABILITY

The n-step probability and steady-state probability are both related to the long-term behaviour of a Markov chain. The n-step probability gives the probability of being in a particular state after n steps, given that the current state is known, while the steady-state probability gives the probability of being in a particular state after many steps. In the context of automotive quality management, a "n-step probability matrix" refers to the matrix of probabilities that describe the probability of the system being in each state after n time steps, given that it starts in a particular state. This can be useful for understanding how the quality of the manufactured parts changes over time and identifying areas where the process can be improved.

The n-step probability matrix can be calculated by raising the transition matrix of the Markov chain to the nth power. The transition matrix, P, represents the probability of transitioning from one state to another in one time step. By raising the transition matrix to the nth power, we can calculate the probability of transitioning from one state to another in n time steps.

Formally, the n-step probability matrix is given by:

P^n = [p_ij^n]

(8)

where p_ij^n is the probability of being in state j after n time steps, given that the system starts in state i.

The n-step probability of a transition from state ‘i’ to state ‘j’. The 1 step transition matrix is as follows:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1s} \\ p_{21} & p_{22} & \cdots & p_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ p_{s1} & p_{s2} & \cdots & p_{ss} \end{pmatrix}$$

(9)

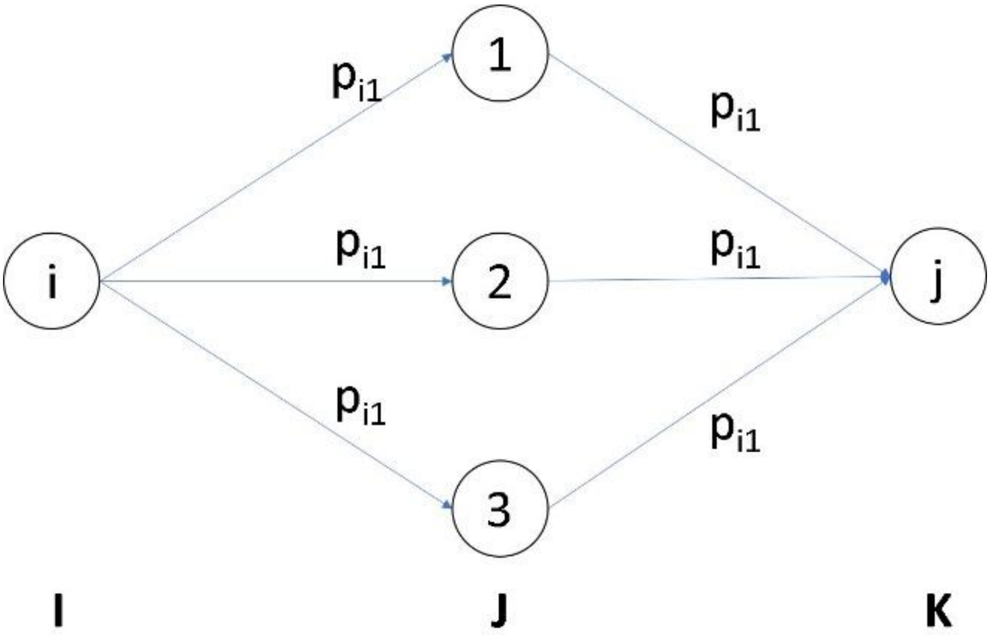


Figure. 3.4 : State diagram with intermittent states

The n step probability here, is used to answer " **If a Markov chain is in the state I at time t, what is the probability that it will be in state j after n periods.**"
Consider the following case studies :

$$P = \begin{bmatrix} & PI & VI & GI & RW1 & RW2 & RT & PS \\ PI & 0 & 0.7 & 0 & 0 & 0 & 0.3 & 0 \\ VI & 0 & 0 & 0.7 & 0.3 & 0 & 0 & 0 \\ GI & 0 & 0 & 0 & 0 & 0.3 & 0 & 0.7 \\ RW1 & 0 & 0.7 & 0 & 0 & 0 & 0.3 & 0 \\ RW2 & 0 & 0 & 0.7 & 0 & 0 & 0.3 & 0 \\ RT & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ PS & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

In the above transition matrix, the raw material from PI takes several steps to reach PS/RT, the states between them are intermittent states.

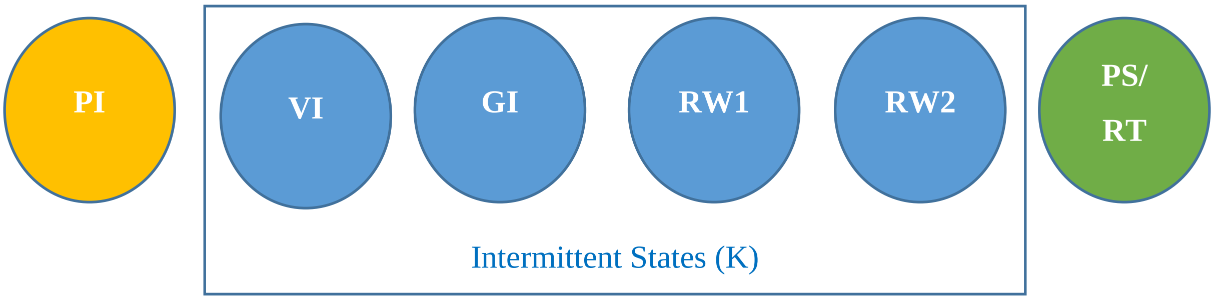


Figure. 3.4: Transient states in Markov Chain

So the intermittent state here are, VI, GI, RW1 and RW2.

So, what is the probability that the raw material would pass V1 and go to GI after 3 materials from now ?

$$\Pr (X= G_1 \mid X0 =Ps) = P_{gi\ vi} \ (PS)$$

$$= \text{element } (PS,PI) \text{ of } P^3$$

For all the three test cases, A and B, the n step probabilities for n=3 and n = 5 are calculated

The results are discussed in section 4.2.2.

3.2.10 STEADY STATE PROBABILITY

In the context of automotive quality management, a "steady-state probability matrix" refers to the matrix of probabilities that describe the long-term behavior of the system, as the number of time steps goes to infinity.

The steady-state probability matrix is given by the limiting matrix of the Markov chain, which represents the long-term probability of being in each state, given that the system starts in a particular state.

Formally, the steady-state probability matrix is given by:

$$L = PL \tag{11}$$

where L is the matrix of steady-state probabilities, and P is the transition matrix of the Markov chain.

In the automotive manufacturing process, the steady-state probability matrix can be used to understand the long-term quality of the manufactured parts, as well as the long-term performance of the process. By analyzing the steady-state probability matrix, one can identify which states are most likely to be occupied in the long-term, and take steps to improve the process in those states.

For example, if the steady-state probability matrix shows that a high proportion of parts are likely to be rejected in the long-term, this could indicate a problem with the process that is causing a high number of defects in the manufactured parts. By identifying this problem, one can take steps to improve the process and reduce the number of defects.

In addition, the steady-state probability matrix can be used to understand how the process is behaving in a steady-state and how it will behave after a long time of operation. This is useful in identifying states that are less visited and taking actions to improve their performance.

For all the test cases, A, B and C, the Limiting matrix are calculated, and the results are discussed in section 4.2.3

3.2.11 ABSORPTION MATRIX

In the context of automotive quality management, an "absorption matrix" refers to a matrix that describes the probability of being absorbed into a particular absorbing state given that the system starts in a particular state. In the context of quality management, an absorbing state is a state that represents the acceptance or rejection of a manufactured part.

The absorption matrix can be calculated by multiplying the fundamental matrix by a vector of the absorption probabilities, which represent the probability of being absorbed into each absorbing state given that the system is in a particular state.

Formally, the absorption matrix is given by:

$$A = F * R \tag{12}$$

where A is the absorption matrix, F is the fundamental matrix, and R is the vector of absorption probabilities.

Then the transition matrix for the absorbing chain may be written as follows:

$$P = \begin{array}{c|c|c} & \begin{matrix} s-m \\ \text{columns} \end{matrix} & \begin{matrix} m \\ \text{columns} \end{matrix} \\ s-m \text{ rows} & Q & R \\ \hline m \text{ rows} & 0 & I \end{array} \tag{13}$$

Here,

I is a m * m identity matrix:

Q depicts transient state transitions as a (s X m) matrix;

R is a (s * m) * m matrix describing transient-to-absorbing state changes;

0 is a zero-matrix m * (s * m).

The fundamental and Limiting matrix for all test cases are calculated and the results are discussed in sections 4.2.4, section 4.2.5, and section 4.2.6.

3.3 REINFORCEMENT LEARNING

Reinforcement learning (RL) is a type of machine learning that is well-suited to problems involving decision-making in uncertain environments. It involves an agent that interacts with an environment and learns to take actions that maximize a reward signal.

3.3.1 MODEL DEVELOPMENT

The measures that need to be taken to optimise the Markov decision process by employing reinforcement algorithms are outlined in this section.

3.3.2.1 DYNAMIC PROGRAMMING

When the model dynamics are known (the probability of transition and the probability of reward), dynamic programming can be used to partition an optimization issue into a number of smaller subproblems and then store the solutions to each of those subproblems. This ensures that each subproblem is only solved once and that the overall issue is optimised.

$$Q(s,a) = P_{ss'}^{a} \times (R_{ss'}^{a} + \gamma Vs) \tag{14}$$

Where $Q(s,a)$= Value of state

$P_{ss'}^{a}$= transition probability moving to next state(s’) from state (s) for action (a)

$R_{ss'}^{a}$ = reward probability for getting into the state (s’) from state (s) for action (a)

γ= discounted factor

Vs = value function

To use dynamic programming to solve the MDP problem, it is necessary to estimate the optimal value function in addition to the optimal policy. When doing value iteration, the value function will initially have random values assigned to it. The numeric values that an agent receives as a reward for successfully completing an activity at a certain state (or states) in the environment are referred to as the reward's value. It is possible that the numerical number will have a positive or negative sign depending on the actions of the agent. In the actual world, we are more concerned with maximising the agent's cumulative reward, which includes all the rewards that the agent receives from the surrounding environment, than we are with optimising the reward that the agent receives from the present circumstance (also called immediate reward). The term "returns" refers to the total amount of the reward that the agent receives from the environment (Bertsekas, 2012).

3.3.2.2 BELLMAN EQUATION

The solution to problems involving reinforcement learning begins with the Bellman equation as the foundational step. The Bellman equation is useful for solving the Markov decision process because it allows us to estimate the optimal policy and value functions. It is possible to find a solution to the bellman equations by employing either value iteration or policy iteration, two distinct types of algorithms. We make advantage of dynamic programming to answer the Bellman problem.

V*(s)= 〖max〗 _π V^π (s) (15)

Where V*(s) = optimum value function

π = policy

$V^\pi (s)$ = value function for policy π

Value iteration

The value function, often known as the V-function and referred to as the state-value function, is a straightforward metric that evaluates how admirable the agent is in any certain condition. The value function is initialised in a haphazard manner, and then, during the process of iteration, the value function is improved and then updated in the value table.

Policy iteration

During the iteration of the policy, the activities that are going to be carried out by the agent must be decided upon or started beforehand. After that, the value table is crafted so that it adheres to the policy. Evaluation of existing policies and development of new ones are both components of the process. Following the completion of the computation of the optimal value function, the following stages are used to estimate the best policy.

1. The policy is initialized randomly.
2. Calculate the value function V(s) for the random policy.
3. If the policy is optimum, stop the process.
4. Else, continue until the policy is improved.

3.3.3 TEMPORAL DIFFERENCE LEARNING

The advantages of both the DP and MC learning algorithms are combined in the Temporal Difference learning method. The value of a state is determined by DP through the application of the bootstrapping approach and the Bellman equation. But to put DP into practise, we need to be familiar with the model dynamics of the environment. On the other hand, MC is a model-free solution, but it cannot be used for jobs that don't occur in episodes. In the TD approach, bootstrapping is used to compute the state value or the Q-value. In contrast to the MC methods, the TD method is applicable to non-episodic jobs (Gosavi, 2019; Ravichandiran, 2018b).

3.3.3.1 TD PREDICTION USING UPDATE RULE

In the context of TD prediction, the estimated new policy is calculated by multiplying the difference between the target and the old estimate by the learning rate and then adding that result to the old estimate. The TD update rule is indeed followed.

$$V(s)=V(s)+\alpha(r+\gamma V(s^\wedge')-V(s)) \quad (16)$$

Where V(s)= new estimate

α = learning rate

r = reward

γ = discounted factor

V(s^')= old estimate

3.3.3.2 TD CONTROL

The objective of TD control is to discover the best policy in the absence of such policy being provided as an input. Because of this, a random policy is used as the starting point, and the best policy is discovered through an iterative process.

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \times maxQ(s',a) - Q(s,a)) \quad (17)$$

Where $Q(s,a)$= value of state

α = learning rate

r = reward

γ = discounted factor

$Q(s',a)$= value of next state

Table 3.2. Reward and Discounted reward

Station		***Reward***	***Discounted Reward***
Package Inspection	**PI**	0.6	0.7
Visual Inspection	**VI**	0.7	0.8
Rework 1	**RW1**	-0.3	0.4
Gauge Inspection	**GI**	0.8	0.9
Rework 2	**RW2**	-0.2	0.2

Pack and Store	**PS**	0.9	0.9
Return	**RT**	-0.4	0.5

Table 3.2 signifies the assumed reward and discounted rewards values for optimum value estimation. The Q(s, a) is computed for each state-action combination, and the maximum value of the Q(s, a) is selected as the answer. This process is repeated for each of the states and the pairings of states and actions that they have, and the results are listed in Table 14. The most effective course of action, as determined by the final q-function, is selected to serve as the basis for the optimal policy.

The findings of the Q(s, a) analysis for the first iteration are presented in Table 3.2. After a number of iterations have been completed, the results of the greatest Q(s, a) are analysed and compared.

3.3.4 CATEGORIZATION OF MATERIAL QUALITY USING TD

A subfield of artificial intelligence known as reinforcement learning, or RL for short, seeks to develop fully autonomous beings that are able to interact with the environments in which they find themselves. They improve their performance over time as a result of going through a process of trial and error while they are being trained, which teaches them the appropriate behaviours to exhibit. (Arulkumaran et al., 2017). Any one of these three approaches may be utilised in order to successfully resolve an RL problem. approaches that use policy search, techniques that use value functions, and a hybrid actor-critic approach that uses both value functions and policy search. Using algorithms that are founded on value functions, one is able to make an educated guess as to the value, or expected mean, of being in each state. In contrast to value functions, policy search approaches do not require the ongoing maintenance of a value function model; rather, they search for the best policy in an unmediated manner. This is in contrast to value functions, which necessitate the maintenance of value function models. Learning strategies that utilise RL algorithms can be categorised as either model-based or model-free strategies. In the field of reinforcement learning, a model is typically defined as the transition dynamic of the environment, the state action pair, and the learning rate. When referring to RL, the phrase "model-free" refers to a scenario in which the agent does not make use of a model or any previous experience in order to attempt to maximise the anticipated reward. This can be contrasted with the situation in which the agent does make use of a model. It does not know what state it will be in after completing a task; the only thing that interests it is the reward that is associated with the state or the action that is associated with the state.

Since the beginning of this decade, the risks that are associated with product recalls have significantly multiplied, which has elevated this kind of risk to the level of being one of the gravest dangers that modern businesses face. In a survey that was conducted not too long ago by Allianz Global Corporate & Specialty (AGCS), it was discovered that the industry that is most negatively impacted by product recalls is the automotive sector. This was followed by the food and beverage industry, which was followed by the information technology and electronics industry. The analysis of product recall claims from 367 insurance companies in a total of 28 countries and 12 different industrial fields was the focus of the research that AGCS conducted between 2012 and the first half of 2017 for the purpose of compiling this report (Allianz, 2017). According to the findings of AGCS, a faulty product or piece of work is the leading cause of product recall claims, followed by product contamination as the second most common cause of these claims. A significant incident will cost more than 12 million United States dollars on average, and the costs associated with the most significant occurrences will be significantly higher than this number. The average cost of a significant incident is more than 12 million dollars. There were ten events that were responsible for more than half of the overall losses that were incurred. A significant problem that, regardless of the industry in which it occurs, undermines the consumers' faith in the product and results in monetary loss for the company is a product recall. The rise in the number of product recalls that have been issued as a result of inadequate product quality is a major cause for concern in the manufacturing industry. This is because of the fact that inadequate product quality has led to more product recalls. An estimate of the material's quality is generated as a result of tracking the path that the material takes while it is being entered for inspection.

The remaining parts of this essay are organised in the following style: The learning strategy of temporal distancing is broken down and discussed in section 2.1. The weights that are estimated based on the timestamps data contained in the core dataset may include contributions from incoming inspection data, although this will vary depending on the component. The development of the Markov decision process (MDP) model is discussed in section 2.2 of this document. In section 2.3, the Q-learning-based temporal difference algorithm that is applied to determine the trajectory is dissected in greater detail. This algorithm is used to determine where the trajectory will go. In Section 3, the findings are dissected into specifics and contrasted with two distinct optimised models.

When modelling the process of estimating the quality of incoming raw materials, the path that the raw material takes while travelling through the incoming inspection path is modelled. The path of the raw material reaches at its conclusion in one of two possible end states: either pack and store or return.

3.3.4.1 TD ALGORITHMS FOR CATEGORIZATION

When it comes to achieving optimum control, one of the strategies that is utilised quite frequently is an algorithm known as the temporal difference algorithm. There are two primary algorithms that are utilised in the process of TD learning. 1) Q-learning, and 2) SARSA, which is an acronym that stands for State Action Reward State Action. On the other hand, Q-Learning is an off-policy approach, while SARSA is an on-policy algorithm. Both of these methods are described as learning systems. The most significant distinction between SARSA and Q-learning is that the Q-values are not necessarily modified based on the size of the reward received in the subsequent state that is received. This is the most important distinction that can be made between the two approaches. For the purpose of the presented case study, Q-learning is chosen instead of SARSA because SARSA learns the secure route, whereas Q–learning learns the optimal way in an accurate manner (Habib, 2019; Ravichandiran, 2018b). Q-learning is also capable of learning the path in a more time and effort efficient manner than SARSA.

The purpose of this section is to demonstrate how Q-learning can be used to estimate the possible next state. Each and every raw material that passes the incoming inspection is distributed to each and every substation so that it can be inspected. The progression of the material is determined by the amount of acceptance and rejection that occurs at each stage. The trajectory is utilised in the process of assigning quality ratings to the material. This objective is accomplished through the utilisation of the Q-learning-based TD algorithm.Q – learning for prediction

$$V(s) = V(s) + \alpha\big(r + \gamma V(s') - V(s)\big) \tag{18}$$

Where V(s)= new estimate

α = learning rate

r = reward

γ = discounted factor

$V(s')$= old estimate

Q- learning – Control or Optimization

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \times maxQ(s',a') - Q(s,a)) \tag{19}$$

$Q(s,a)$ = value of current state

α = learning rate

r = reward

γ = discount rate

$Q(s',a')$= value of next state

Figure 3.5 represents the flowchart of the Q-learning. During the estimation, the value of the following state V' (s) is initialized to zero.

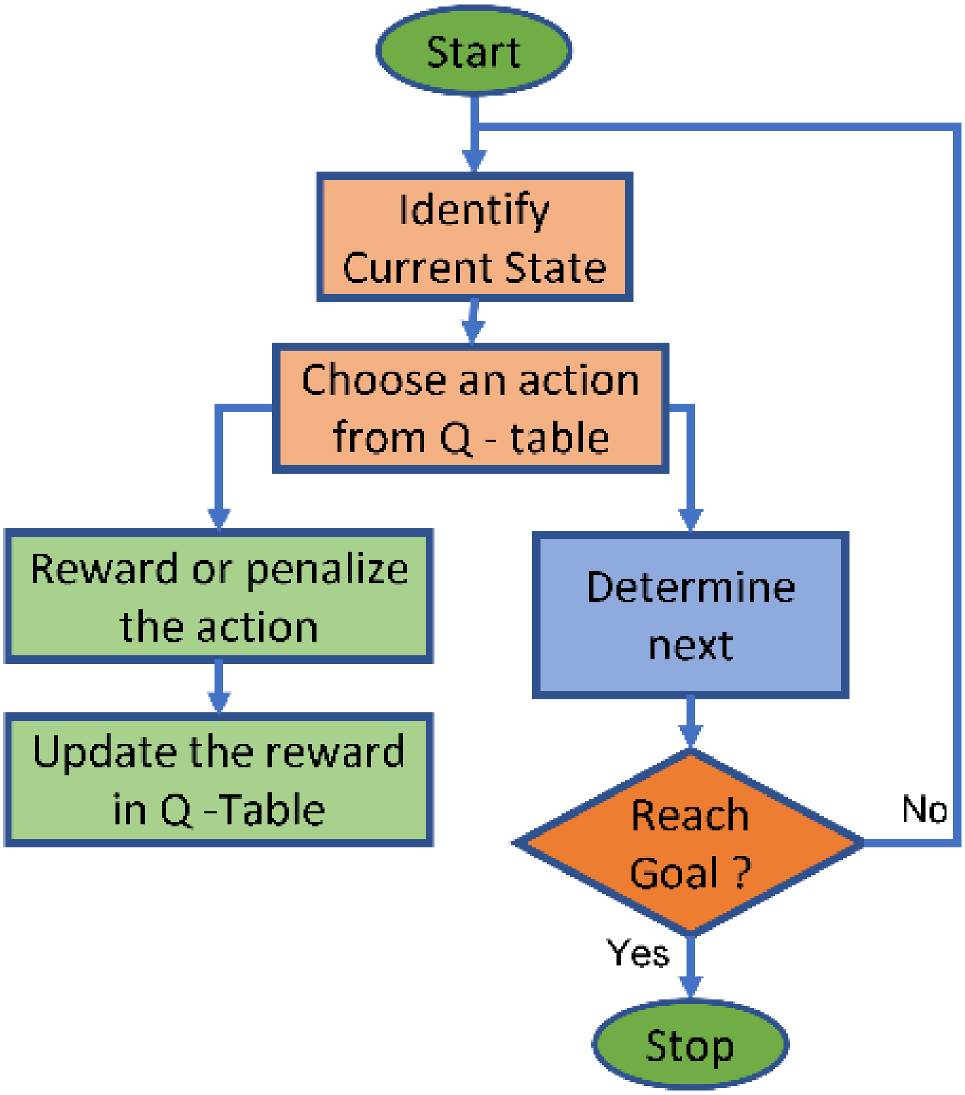

Start
Identify Current State
Choose an action from Q - table
Reward or penalize the action
Determine next
Update the reward in Q -Table
Reach Goal ?
No
Yes
Stop

Figure 3.5. Flowchart of Q-learning algorithm

When attempting to estimate the value of the state, the learning rate or step size will be used to determine whether or not the estimation will converge. As a result of this, the learning rate () has been calibrated to 0.1 in order to guarantee a seamless progression from one state to the next. If the agent chooses the appropriate action, they will receive points as a reward; however, if they choose the incorrect action, they will receive points as a punishment. The value of the rewards, denoted by the letter r, and the value of the penalties, denoted by the letter p, are both represented as integers, with the rewards having a positive value and the penalties having a negative value. The term "discount factor," which is symbolised by the Greek letter "," refers to a concept that evaluates the magnitude of immediate benefits in relation to those that will be obtained in the future. The discount factor, as well as the reward and penalty for each step, are tabulated below in Table 2, which can be found further down this page. Both the size of the reward and the size of the discount were chosen at random and then implemented into the system.

Table 3.3. Moves, reward, and discount factors.

Moves	**Reward**	**Discount Factor**
PI-VI	5	0.7
Pi-RT	-3	0.1
VI-GI	5	0.8
VI-RW1	-2	0.4
GI-PS	3	0.8
GI-RW2	-4	0.4
RW1-VI	1	0.6
RW1-RT	-2	0.1
RW2-GI	0.5	0.6
RW2-RT	-3	0.1

The numeric values that an agent receives as a reward for successfully completing an activity at a certain state (or states) in the environment are referred to as the reward's value. This is because the agent receives these values as a reward for successfully completing the activity. Depending on what the agent does, the numerical value could end up having a positive or a negative sign attached to it. This is a possibility. In the real world, we are more concerned with maximising the cumulative reward, which refers to all of the rewards the agent receives from the environment, as opposed to the reward that the agent receives from the present circumstance. This is because the cumulative reward takes into account all the rewards that the agent receives from the environment (also called immediate reward). Returns is a term that refers to the total amount of reward that an agent receives from their environment. Returns can be positive or negative.

3.4 DEEP ACTIVE LEARNING AND COMPUTER VISION BASED VISUAL INSPECTION

This section details the experimental design devised to implement the Deep Active Learning (DAL) approach to optimize the Visual inspection process.

The Quality Inspection is carried out using the following methods: 1) Radiography, 2) Scanning Laser system, 3) Visual Inspection.

In manufacturing process, the visual inspection is an important form of quality inspection. The non-conformances identified at the early stage can save considerable amount of process and maintains product quality.

Visual inspection research has a long history spanning the 20th century and continuing to the present day (See, 2012).. See presented a detailed review on Visual Inspection (See, 2012). In his report he highlighted, the various visual inspection techniques between 1950 and 2012. The report details that in rare instances, failure to spot defects might result in severe repercussions, ranging from bodily harm to even death. In the field of aviation maintenance and inspection, 65 injuries in 1988, 111 fatalities in 1989 and 2 lives lost in 1996 were reported due to the defects that were unnoticed during the inspection.

In other circumstances, inspection errors might not result in physical harm or loss of life, but they might nevertheless directly incur expenses for the business. On the one hand, it is possible to supply a faulty product, which can have a detrimental effect on the level of satisfaction experienced by the customer as well as the chance of future business. On the other hand, a good item could be deemed defective, in which case it would need to be reworked or trashed, which would result in extra expenses for materials and labour.

The automation of inspection started to raise during 1990s while the first automated inspection dates to 1983. Drury and Watson outlined the good practices in visual inspection (Drury & Watson, 2002). A detailed survey on the automated visual inspection is described by Huang and Pan (S. H. Huang & Pan, 2015). Automated Visual Inspection as discussed has been widely researched. While new technologies and sensors are introduced to improve the detection accuracy and reduce the human cost, computer vision remains the promising solution in many of the manufacturing industries. Computer Vision techniques for detect detection in manufacturing is explained by Zhou and his associates (Zhou et al., 2022)

In the industrial sector, human visual inspection abilities are still preferable to automated testing for product quality and standard compliance (T. L. Johnson et al., 2019). The cost of manual visual inspection remains a costly venture due to need for a well-trained individual. Defect detection, quality monitoring plays an important role in the manufacturing product cycle.

To demonstrate the visual inspection using computer vision in the industrial setup, the publicly available dataset for casting product for quality inspection is chosen. A brief description of the casting dataset is explained in Section 3.4.1 of this chapter. An extended description of the casting dataset is presented in Appendix A.

3.4.1 DATASET: IMPELLER QUALITY INSPECTION

The Images Classification is the popular task in computer vision. With the advent of Internet of things and the availability of abundant images on the internet, it is not difficult to try our any clean dataset used for academic research. However, this research focuses on the problems that are present in the manufacturing industry. Hence a suitable industrial dataset is required. As a fact, industrial data is a proprietary and are seldom available for public research. Nevertheless, we identified an industrial dataset to support the quality inspection case study.

In the manufacturing sector, maximizing profits requires a significant reduction in the number of processing errors that occur during the manufacturing process. It is necessary to secure a budget for quality assurance, put in place manual inspection work, and conduct a thorough review of the manufacturing process to cut down on the number of processing errors. Especially, the inspection process is carried out by a lot of different businesses, but this results in issues such as inconsistent accuracy, which is dependent on inspection workforce, and increasing expenses associated with labour. Casting is a process that involves pouring molten metal into a mould and then shaping the metal into the desired form after it has cooled. The following is a summary of some of the defects that occurred throughout the casting process. Defects due to shrinkage, mould materials, pouring metal, metallurgical, blow holes, pinholes, and burr to name a few. In the casting industry, defects are something that should be avoided at all costs.

Every industry has a quality inspection department that is responsible for the removal of faulty products. However, the most significant issue is that this examination procedure is performed by hand. It is a very time-consuming process, and because of the inherent fallibility of humans, the results cannot be relied upon to be entirely correct. This may be due to the entire order being declined for whatever reason. Therefore, it results in a significant loss for the organisation.

The dataset used in this research was published in Kaggle by (Dhabi, 2019) owners of the casting industry Pilot Techno Cast. The dataset is distributed under the creative commons license.

The owners came up with an idea to automate the inspection procedure, and as a result, the images were captured for the Impeller units were captured. The dataset provides images of impellers for submersible propellors.

Figure 3.4 (a) Submersible Pump

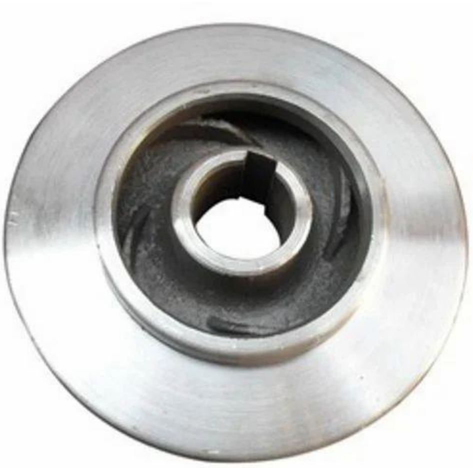


Figure 3.4 (b) Impeller

This dataset is collected under stable lighting environment with extra arrangements. The **canon EOS 1300D DSLR** camera is used to collect the data. All the images are converted into a fixed size of (300*300) grayscale.

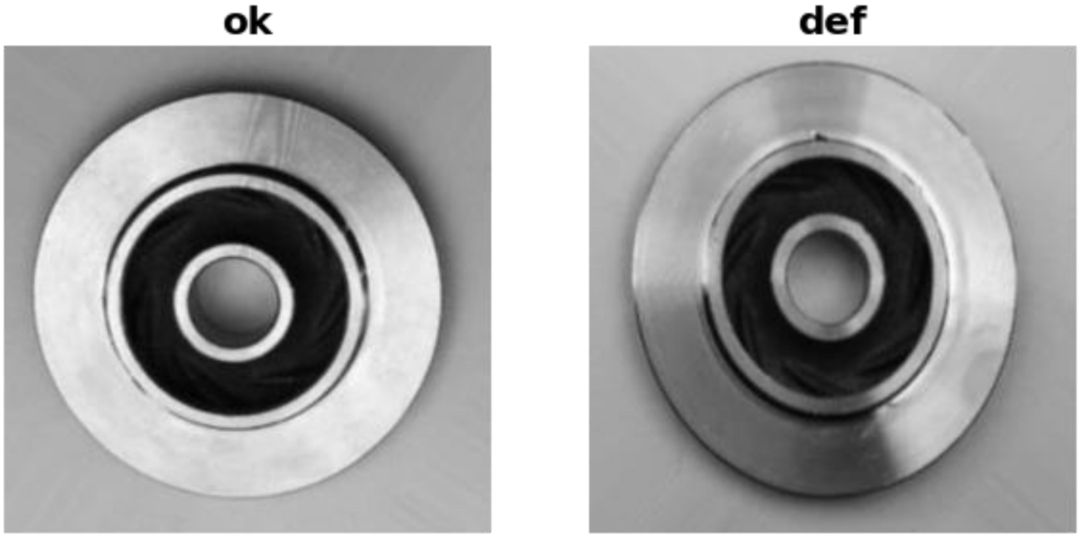


Fig 3.5 OK and Defective images in Impeller samples

The image data is labelled with **ok(normal)** and **def(defect/anomaly)** in advance. In addition, since it is necessary to illuminate the image in a stable condition when acquiring the image, the data was acquired based on a special lighting setting. In the casting dataset investigated in this research, the authors have already applied a variety of pre-processing techniques for the purpose of removing irrelevant noise from photographs. These photos of the casting were obtained using certain setups that guaranteed the image would be obtained under consistent lighting circumstances. Nevertheless, in a manufacturing environment that is representative of the real world, the lighting condition obviously shifts over the course of time, which might result in classification mistakes if the vision system depends on constant lighting conditions. The above condition is also addressed by Nguyen and his team who have used the same dataset to build a defect classification model (H. T. Nguyen et al., 2020).

Data preparation

The dataset contains images from two classes “ok_front” and “defective front”. The total number of images in test dataset is **6633** which combines both the classes. The test dataset contains **715** images from the two classes.

The dataset is imbalanced as the number of images for the “ok” class is 3141 while the number of images for the “defect” class is 4217.

Table 3.1 Casting Dataset Description

	Train		**Test**		**Total**
	ok_front	**def_front**	**ok_front**	**def_front**	
casting 300 x 300	3759	2878	263	458	7358
casting 512 x 512	781	519	-	-	1300

Previous Research on the same dataset

Nguyen and Shin (H. T. Nguyen et al., 2020) applied transfer learning technique to retrain the casting dataset on the VGGNet, ResNet, DenseNet and GoogleNet pretrained models. They further tested the trained models by deploying them on the Jetson Nano Edge device. Their study proved that ResNet took the least time for training (65 seconds to train 1 epoch) while performing on the highest FPS (frames per second) during inference. This leads to our choice of selecting ResNet as the go to model for our research work.

Kim and his team (Kim et al., 2022a) proposed a CNN based classification of the defect detection on the same casting dataset chosen for this research. They examined the sample size sensitivity of a manufacturing picture dataset, thought about the number of samples needed to stabilise a smart factory, and compared the performance of the model according to different sample sizes.

Active Learning Training Scheme

The CNN models were trained using Adam optimizer [Knigma et al, 2014]. The learning rate is varied between the range of 0.001 to 0.1. at the step size of 50.

Computing environment

The algorithms and the Active Learning approach were implemented using PyTorch library [Paszke et al, 2019]. The experiments were run on Cloud infrastructure with a single NVIDIA GeForce RTX of 64 GB memory.

3.4.2 IMAGE CLASSIFICATION USING TRADITIONAL DEEP LEARNING (PASSIVE LEARNING)

ConvNet or Convolutional Neural Networks (CNN) was a ground-breaking

Traditional CNNs have a severe flaw in that they must learn the complete feature map, which necessitates the use of many parameters. This therefore implies that they are slower runners and relatively expensive to train.

A family of neural networks called ResNets was put out as a replacement for conventional CNNs. ResNets in particular make advantage of skip connections enabling them to be considerably smaller than conventional CNNs while still achieving comparable performance. Any neural network design may employ skip connections, but convolutional neural networks benefit from them the most since they allow you to reuse portions of your feature map between layers in various spots.

Restnet18 architecture

The residual neural network (or ResNet) is introduced by He, Zhang and others (He et al., 2015) to address the vanishing gradient issues that arose when the depth of the deep neural networks increased. There are many variants of Resnet such as 18, 34, 50 and 101.

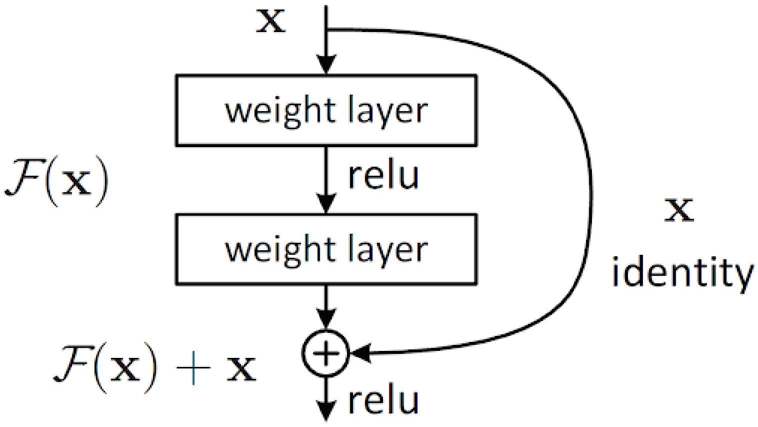

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig 3.6 The Residual Block of the Deep Residual Network Architecture ()

ResNet Models are perfect in situations when great classification accuracy is required. Authors from the literature have admitted that the choice of the backbone architecture for Active Learning achieves different classification accuracy, with ResNet-18 achieving the highest. ResNet models offer reasonable model sizes and extremely high accuracy and are the most influential models in the field of computer vision. To further understand about the deep residual networks, Shafiq and Gu have presented a detailed survey on Deep Residual learning (Shafiq & Gu, 2022a).

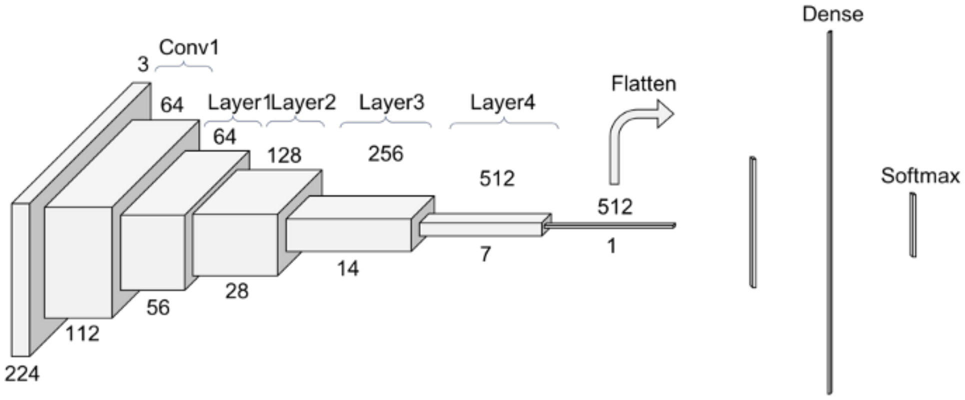


Figure 3.7: Resnet Architecture ()

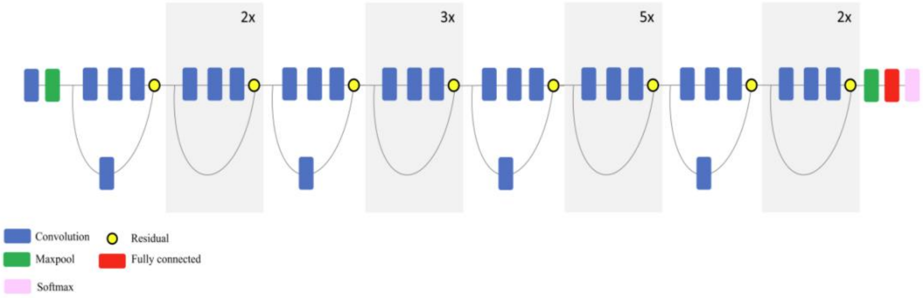


Figure 3.8: Schematic of the ResNet architecture (Mahdianpari et al., 2018)

3.4.3 DEEP ACTIVE LEARNING

According to NVIDIA, if people were to label the data for a 100-car fleet that drove for eight hours each day, they would require more than one million labellers to do it. To perform only 20 percent better than a person, it takes autonomous cars approximately 11 billion kilometres of driving experience(Ram Sagar, 2020).

Active Learning is a widely researched field over the past decades. Model optimization is the fundamental goal of any deep learning.

Deep Active Learning is a promising technique which reduces the human effort required for data annotation. Zhan and his associates presented a detailed survey on the Deep Active learning techniques (Zhan, Wang, et al., 2022b).

Deep active learning (AL) approaches may be broken down into three distinct categories: learning-based methods, uncertainty-based methods, and representation-based methods. In addition, some ways have suggested using a combination of the two approaches.

Uncertainty-based approaches look for examples that are challenging to memorise to locate them. Both Bayesian [8, 15, 16, 23] and non-Bayesian techniques [25, 32] have been used in the development of several methodologies for estimating the degree of uncertainty associated with neural networks. Estimation of the posterior uncertainty was proposed by Gal et al. (Gal et al., 2017) through the use of dropout for active learning. The entropy of the SoftMax output in a neural network was utilised as a proxy uncertainty measure to query samples and is discussed in Wang et al (K. Wang et al., 2017) 's study. The ensemble technique was used by Beluch et al. (Beluch et al., 2018) to quantify the uncertainty of the forecast, and the selection of fresh samples was based on a statistical measure of the committee's disagreement known as the variation ratio

(E. H. Johnson & Freeman, 1966). They demonstrate that this strategy works far better than any other uncertainty-based strategies.

The goal of representation-based approaches [36, 43], which are also known as density-based methods, is to choose a varied group of samples that most accurately depicts the distribution of the whole dataset. Sener et al. (Sener & Savarese, 2018) developed a formulation for the active learning issue known as core-set selection and demonstrated its usefulness for CNNs.

Learning-based methods [39, 44] use the utilisation of an auxiliary network module and loss function in order to learn a measurement of the information gain from new samples. Yoo et al. [44] suggested learning a loss prediction module to anticipate target losses of unlabelled samples and picks samples with the greatest expected loss. This would include selecting samples with the highest predicted loss. It is also possible to look at it from the perspective of a pseudo uncertainty heuristic. Sinha et al. [39] introduced a method to semi supervised active learning that learns a VAE-GAN hybrid network to pick unlabelled samples that are not well represented in the labelled set. This technique utilises semi supervised active learning. It is also possible to think of it as a means of representing something.

The figure below depicts the deep active learning pipeline. Ranganathan proposed a

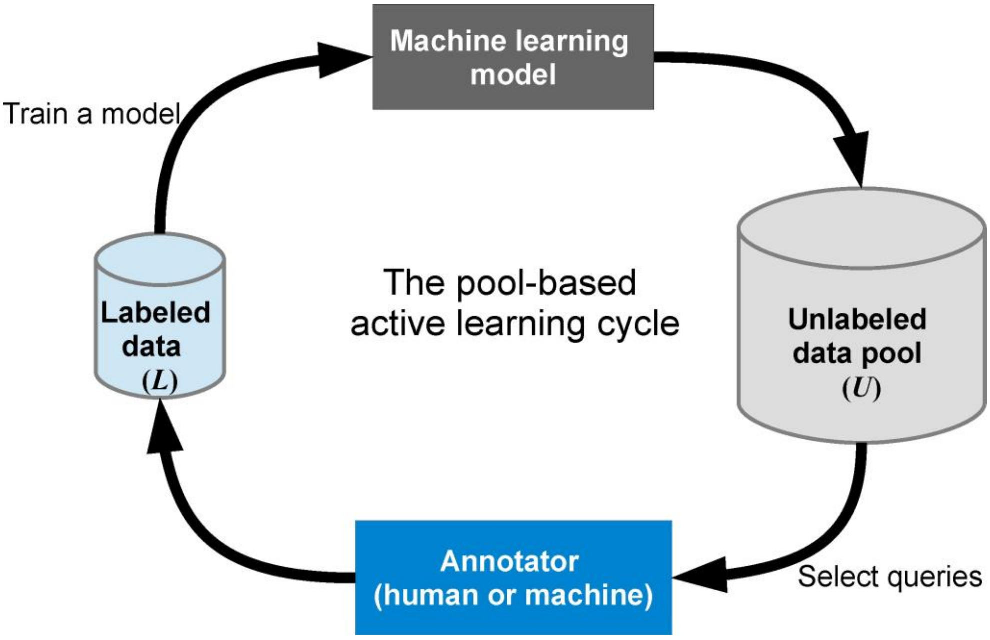

Machine learning model
Train a model
Labeled data (*L*)
The pool-based active learning cycle
Unlabeled data pool (*U*)
Annotator (human or machine)
Select queries

Figure 3.9: Pool based scenario pipeline (Kale, 2018)

Pytorch is utilised in the implementation of the ResNet18 Convolutional Neural Networks of various complexity.

3.4.2.1 ENTROPY SAMPLING

The uncertainty of the samples is evaluated using this approach by calculating their entropy value, which is denoted by (Eni). A larger Eni number indicates that the sample I has a greater degree of uncertainty, which may be described as follows:

The entropy sampling (Shannon, 1948) approach has been utilised as a method of uncertainty measurement in several earlier research that have been conducted on the AL domain (J. Chen et al., 2006; M. Tang et al., 2002). Entropy sampling refers to the situation in which the learner decides to question the labels of the samples that have the greatest amount of entropy in terms of class prediction information. The following is the formula used to compute entropy:

$$H(x) = -P\ (y|x)\log(P\ (y|x))$$

$$y \in Y$$

where P (y|x) is the posteriori probability, H is the uncertainty measurement function and Y = {y1, y2.., yk} [67].

Entropy sampling is a query strategy that can be used in active learning to select the most uncertain samples for labelling. The steps for using entropy sampling typically involve:

1. Define the probabilistic model: This is typically a deep neural network that will be used for the active learning process. The model is initially trained on a small set of labelled data.
2. Estimate model uncertainty: The model is used to make predictions on a pool of unlabelled data, and the model's uncertainty is estimated for each sample in the pool. This is typically done by measuring the entropy of the model's predictions, where high entropy corresponds to high uncertainty.
3. Select the most uncertain samples: The samples with the highest entropy are selected to be labelled and added to the training set.

4. Label the selected samples: The selected samples are then labelled by an oracle, such as a human annotator.
5. Re-train the model: The model is re-trained on the updated labelled data.
6. Repeat the process: The process is repeated multiple times, with the model's uncertainty being re-estimated on the pool of unlabelled data after each re-training.

Entropy sampling is a simple and effective query strategy that can be used in a variety of active learning tasks, especially when the number of classes is small.

3.4.2.2 RANDOM SAMPLING

The approach known as random selection (RS) is typically utilised to provide as a point of reference for evaluating active learning. In this approach, the unlabelled samples for each cycle are chosen at random using a random number generator.

The random sampling approach describes the situation in which the learner makes the decision to query the label for a sample at random. This approach is frequently used as a benchmark against which the performance of other methods may be compared and evaluated. (T. Luo et al., 2005).

Random sampling is a query strategy that can be used in active learning to select samples for labelling at random. The steps for using random sampling typically involve:

1. Define the probabilistic model: This is typically a deep neural network that will be used for the active learning process. The model is initially trained on a small set of labelled data.
2. Select random samples: Randomly select a subset of samples from the pool of unlabelled data to be labelled and added to the training set.
3. Label the selected samples: The selected samples are then labelled by an oracle, such as a human annotator.
4. Re-train the model: The model is re-trained on the updated labelled data.
5. Repeat the process: The process is repeated multiple times, with random samples being selected for labelling and the model being re-trained on the updated labelled data.

6. Stop: The process is stopped when a certain stopping criterion is met, such as a certain level of model performance, a maximum number of queries, or a budget for the number of allowed queries.

Random sampling is a simple query strategy that can be used in a variety of active learning tasks. However, it is not as effective as other query strategies, such as uncertainty sampling or query-by-committee, which tend to select samples that are more informative for learning.

3.4.2.4 DEEP BAYESIAN ACTIVE LEARNING (DBAL) SAMPLING

Deep Bayesian active learning algorithms are a type of machine learning algorithm that actively selects the most informative data samples for learning. These algorithms typically involve the following steps:

Define the probabilistic model: This is typically a deep neural network that will be used for the active learning process. The model is initially trained on a small set of labelled data.

Estimate model uncertainty: The model is used to make predictions on a pool of unlabelled data. The model's uncertainty is then estimated for each sample in the pool.

The deep Bayesian active learning algorithm would generally have the following steps:

1. Start: Initialize the probabilistic model (e.g., deep neural network) with a small set of labelled data.
2. Estimate Uncertainty: Use the model to make predictions on a pool of unlabelled data and estimate the model's uncertainty for each sample.
3. Select Samples: Select the samples with the highest uncertainty using a query strategy (e.g., uncertainty sampling).
4. Label Samples: Label the selected samples by an oracle (e.g., human annotator)
5. Re-train: Re-train the model on the updated labelled data.
6. Repeat: Repeat the process of uncertainty estimation, sample selection, and re-training multiple times.

7. Stop: Stop the process when a certain stopping criterion is met (e.g., a certain level of model performance, a maximum number of queries, or a budget for the number of allowed queries).
8. Test: Test the performance of the final model on a held-out test set.

The main advantage of this approach is that it allows to reduce the amount of labelled data needed for training, which can be useful when labelled data is scarce or expensive to obtain. Additionally, using active learning can also lead to faster convergence of the model's performance, as the algorithm is able to focus on the most informative samples.

3.4.2.5 BAYESIAN ACTIVE LEARNING BY DISCRIMINATION (BALD) SAMPLING

Bayesian active learning by discrimination (BALD) is a specific active learning strategy that uses the Bayesian formulation of a neural network to select the most informative samples for learning. BALD typically involves the following steps:

1. Define the probabilistic model: This is typically a deep neural network that will be used for the active learning process. The model is initially trained on a small set of labelled data.
2. Estimate model uncertainty: The model's uncertainty is estimated for each sample in the pool of unlabelled data. BALD uses the mutual information between the model's predictions and the model's parameters to estimate the uncertainty.
3. Select the most informative samples: The samples with the highest expected reduction in the model's uncertainty are selected to be labelled and added to the training set.
4. Label the selected samples: The selected samples are then labelled by an oracle, such as a human annotator.
5. Re-train the model: The model is re-trained on the updated labelled data.
6. Repeat the process: The process is repeated multiple times, with the model's uncertainty being re-estimated on the pool of unlabelled data after each re-training.

7. Stop: The process is stopped when a certain stopping criterion is met, such as a certain level of model performance, a maximum number of queries, or a budget for the number of allowed queries.

BALD is particularly useful when data is scarce, as it is able to select the most informative samples even when the pool of unlabelled data is large. Additionally, BALD can also be useful when the cost of labelling data is high, as it allows to reduce the number of samples that need to be labelled.

3.4.4 DEEP ACTIVE LEARNING FOR CASTING DATASET

The deep active learning approach is applied on the casting dataset with the four query sampling algorithms namely: Entropy, Random, DBAL and BALD. The choice of these four query sampling algorithms were made based on the literature (Mayumu, 2022; Ren et al., 2022; Settles, 2009; Shafiq & Gu, 2022b; Zhan, Wang, et al., 2022c) The classification model is trained by adapting the Pytorch framework developed by Munjal et al (Munjal et al., 2020, 2022) and the implementation by (Chandra & Balasubramanian, 2021).

The implementation follows the training parameters and guidelines suggested in the literature (Adrian et al., 2021; Jo et al., 2022; Munjal et al., 2020, 2022).

Table 3.x Modal training parameters used for the casting dataset.

Parameter	**Values**
Network Architecture	Resnet18
Loss function	Cross entropy
Query sampling algorithms	Entropy, random, DBAL and BALD
Budget Size	300
Initial Learning rate	0.01
Passive learning iterations	1
Active learning episodes	5
Optimizer	SGD

Batch size	32
Image size	300 x 300
Number of classes	2

The details of the usage of the code are presented in Appendix A of this thesis. The results of the model training are presented in Section 4.4 and the findings are discussed subsequently.

3.5 CHAPTER SUMMARY

This chapter detailed the study's three components. The Markov decision technique and RFID tag data incoming inspection case study are described in 3.2. State machine diagrams clarify the case study. The inbound inspection case study and substations are detailed. This section shows how to compute Markov chain characteristics for experimental instances. This section also describes the transition matrix, transition probability, and techniques used to create the raw material acceptance prediction model. In Part 3.3, reinforcement learning optimisation for the prediction model created using the Markov decision process solved in the previous parts were presented. This part next discussed how to solve the raw material acceptance prediction model using the temporal difference RL algorithm and dynamic programming. This section described how to use temporal difference to develop a model for material quality categorization estimation. Section 3.4 described the impeller defects casting dataset and deep learning visual inspection. This section discusses how to apply deep active learning to optimise model building. The subsequent chapter showcases the results for the experiments that are described in this chapter.

CHAPTER 4

CHAPTER FOUR: RESULTS AND DISCUSSION

4.1 CHAPTER INTRODUCTION

The results of the optimization models are presented in Chapter 4. This chapter also compares the results to previous research and concludes by providing a summary of the findings. The chapter is divided up into sections numbered 1, 2, and 3. The findings of the raw material acceptance prediction model for the arriving inspection case study utilising the Markov decision process are presented in Section 4.2. The findings are then analysed in light of the Markov qualities and contrasted with the relevant research in the field. The findings of the optimised models for the acceptance of raw materials are presented in Section 4.3. These models make use of dynamic programming and temporal difference. The results are compared to one another, and then an in-depth discussion follows. In addition, the findings of the model used to estimate the material's quality are provided and analysed in the appropriate context. The outcomes of the visual inspection model carried out with the assistance of deep learning computer vision models are presented in Section 4.4. The results obtained through the deep learning technique that has been used traditionally as well as the deep active learning approach have been built on and analysed. In conclusion, section 4.5 provides a synopsis of the findings discussed throughout the chapter.

4.2 INCOMING INSPECTION USING MARKOV DECISION PROCESS

The incoming inspection case study is formulated into a Markov Decision process. The following Markov properties are computed to study the behavior of the process and the results are discussed accordingly.

1. N Step probability
2. Steady State probability
3. Absorbing matrix
    a. Fundamental matrix
    b. Limiting matrix

4.2.1 RESULTS OF THE N STEP PROBABILITY ESTIMATION

In a Markov chain, the n-step transition probability gives the probability of transitioning from one state to another after n steps. It is the probability of being in a particular state after n steps, given that the current state is known.

Consider the transition probability values of 70/30 shown in Table 4.1. The results of the N step probability were computed for the third (n=3) and fifth (n=5) incoming raw material from the present time 't'. The event of interest is to go from state 'i' to 'j' in n-steps. The intermediate states here are Visual Inspection (VI), Gauge Inspection (GI) and Rework (RW).

Research Question: What is the probability of acceptance when the nth raw material moves from state 'i' to state 'j'

Consider two cases A and B where 'i' = GI and 'j' = PS

Case A: n-step probability estimation for n =3

Table 4.3 n-step probability for n=3

i, j	**PI**	**VI**	**GI**	**RW1**	**RW2**	**RT**	**PS**
PI	0	0.147	0	0	0.147	0.363	0.343
VI	0	0	0.294	0.063	0	0.153	0.49
GI	0	0	0	0	0.063	0.09	**0.847**
RW1	0	0.147	0	0	0.147	0.363	0.343
RW2	0	0	0.147	0	0	0.363	0.49
RT	0	0	0	0	0	1	0
PS	0	0	0	0	0	0	1

For the above case where n =3 presented in Table 4.3, the probability that the 3rd raw material being accepted and moves from GI to PS is **84.7%.**

Case B: n-step probability estimation for n =5

Table 4.4 n-step probability for n =5

Power (3)	**PI**	**VI**	**GI**	**RW1**	**RW2**	**RT**	**PS**
PI	0	0.031	0	0	0.062	0.420	0.487
VI	0	0	0.092	0.013	0	0.198	0.696

GI	0	0	0	0	0.013	0.109	**0.878**
RW1	0	0.031	0	0	0.062	0.420	0.487
RW2	0	0	0.031	0	0	0.376	0.593
RT	0	0	0	0	0	1	0
PS	0	0	0	0	0	0	1

Similarly, for the above case where n =5 presented in Table 4.4, the probability that the 5th raw material being accepted and moves from GI to PS is **87.8%**.

4.2.1.1. DISCUSSION ON THE N STEP PROBABILITY

The results for the N step probability are presented above in Table 4.2 and 4.3 shows. From the results for the cases A and B, the n-step transition probability can be used to analyze the long-term behavior of a Markov chain. The probability of being in a particular state after many steps, called the steady-state probability, can be found by taking the limit of the n-step transition probability as ‘n’ goes to infinity.

For the chosen transition probability of 70/30, the raw material being accepted for Gauge Inspection has improved from **84.7** to **87.8**. This is an increase of 31% between two steps.

In the automotive industry, the n-step probability can be used in quality management to analyze the probability of a product or process being in a particular state after a specific number of steps or a specific period of time.

For example, the n-step probability can be used to analyze the performance of a production line over time, identifying which processes or equipment are causing the most defects and determining the proportion of defective products at a specific point in time. This information can be used to improve the design of the production line, implement quality control measures, and optimize the use of resources.

It can also be used to evaluate the effectiveness of corrective actions taken to improve quality, such as changes to the production process, training of workers, or maintenance of equipment. By analyzing the n-step probability, one can understand how the changes are impacting the quality of the product over specific time intervals.

Additionally, in the automotive industry, the n-step probability can also be used to model the reliability of the products, which means modeling the probability of failure of a component/system over time, and the n-step probability of the Markov Chain can be used to understand the reliability of the product at specific time intervals.

In conclusion, the n-step probability can be a valuable tool in quality management in the automotive industry, helping to identify and mitigate sources of defects, and improve overall product quality and reliability over specific time intervals.

4.2.2. RESULTS OF THE STEADY STATE PROBABILITY

The n-step probability can be used to find the probability of certain sequences of states occurring in the Markov chain, while the steady-state probability can be used to understand the long-term behavior of the system.

The relation between the two probabilities is that as the number of steps (n) increases, the n-step probability tends to approach the steady-state probability. This is because as n becomes larger and larger, the system is less influenced by the initial state, and the behavior of the system becomes more predictable.

The stationary distribution of Markov Chains is when the probability remains unchanged after a point of time. This is calculated using n-step transition probability. The sum of the probabilities turns to 1.

Consider for the cases A, B and C probability transition matrix discussed above, the steady state probabilities are found using the power method and the results are furnished below.

Table 4.3 Stationary distribution

Case	N-steps	Return (RT) probability	Pack and Store (PS) probability
A	21	0.7727	0.2273
B	18	0.4563	0.5437
C	13	0.1946	0.8054

In reference to Table 4.3, after 21st step, the transition probability matrix attains a state where the probability ceases at certain value, RT at 0.7727 and PS at 0.2273. Similarly for the other test cases B, C, the values cease at 18th and 13th step, respectively.

4.2.2.1. DISCUSSION ON THE STEADY STATE PROBABILITY

In the automotive industry, steady-state probability can be used to analyze the long-term behavior of a quality management system. In particular, it can be used to understand the probability of a product or process being in a particular state, such as conforming to specifications or being defective.

For example, the steady-state probability can be used to analyze the performance of a production line, identifying which processes or equipment are causing the most defects and determining the proportion of defective products in the overall output. This information can be used to improve the design of the production line, implement quality control measures, and optimize the use of resources.

It can also be used to evaluate the effectiveness of corrective actions taken to improve quality, such as changes to the production process, training of workers, or maintenance of equipment.

Additionally, in the automotive industry, Markov Chain also used to model the reliability of the products, which means modeling the probability of failure of a component/system over time, and the steady-state probability of the Markov Chain can be used to understand the long-term reliability of the product.

Overall, the steady-state probability can be a powerful tool in quality management in the automotive industry, helping to identify and mitigate sources of defects and improve overall product quality and reliability.

4.2.3. RESULTS OF THE ABSORBING MATRIX

In the automotive industry, an absorbing Markov chain can be used to model the reliability of a system, where the states represent different failure modes. The absorbing state represents the final failure state, and the transient states represent the different modes of operation before the final failure. The absorption probability can be used to understand the probability of reaching the final failure state from a particular mode of operation.

4.2.5. RESULTS OF THE FUNDAMENTAL MATRIX

In the automotive industry, quality inspection of the manufactured parts is an important step to ensure that the final product meets the desired quality standards.

The fundamental matrix calculates the estimated number of times the system will be in each condition until reaching an absorbing state, which is either accepted or rejected.

Table 4.4 Transition probabilities of the three cases

	CASE A				
	PI	**VI**	**GI**	**RW1**	**RW2**
PI	1.0	0.7	0.4	0.3	0.2
VI	0.0	1.3	0.9	0.7	0.4
GI	0.0	0.0	1.3	0.0	0.7
RW1	0.0	0.7	0.4	1.3	0.2
RW2	0.0	0.0	0.7	0.0	1.3

	CASE B				
	PI	**VI**	**GI**	**RW1**	**RW2**
PI	1.0	0.9	0.7	0.4	0.1
VI	0.0	1.3	0.9	0.5	0.2
GI	0.0	0.0	1.2	0.0	0.2
RW1	0.0	0.8	0.6	1.3	0.1
RW2	0.0	0.0	1.0	0.0	1.2

	CASE C				
	PI	**VI**	**GI**	**RW1**	**RW2**
PI	1.0	0.4	0.2	0.2	0.2
VI	0.0	1.3	0.6	0.8	0.5

GI	0.0	0.0	1.2	0.0	1.0
RW1	0.0	0.5	0.3	1.3	0.2
RW2	0.0	0.0	0.2	0.0	1.2

For instance, in case study A, the probability of raw material would visit RWI from VI before being absorbed at PS is 70%. Similarly, In case B, there is 40 % chances that the raw material would be sent to rework, in spite of qualifying at packaging Inspection (PI). Interestingly, a raw material, which has been reworked for visual inspection defects has 20% chances of being sent to rework station (RW2) for dimensional nonconformities.

4.2.5.1. DISCUSSION ON RESULTS FOR FUNDAMENTAL MATRIX

The fundamental matrix of a Markov chain can be used to analyze the long-term behavior of the quality inspection process, which can be useful in understanding the performance of the inspection system and identifying potential areas for improvement. In the automotive industry, one use of the fundamental matrix is to calculate the expected number of times that the system will be in each state before it reaches an absorbing state, which represents a part that is either accepted or rejected. This information can be used to understand how often parts are inspected multiple times and how often parts are accepted or rejected on the first inspection.

Additionally, the fundamental matrix can be used to calculate the expected time to absorption, which represents the expected time it takes for a part to be accepted or rejected. This information can be used to understand the efficiency of the inspection process and identify bottlenecks that are causing delays.

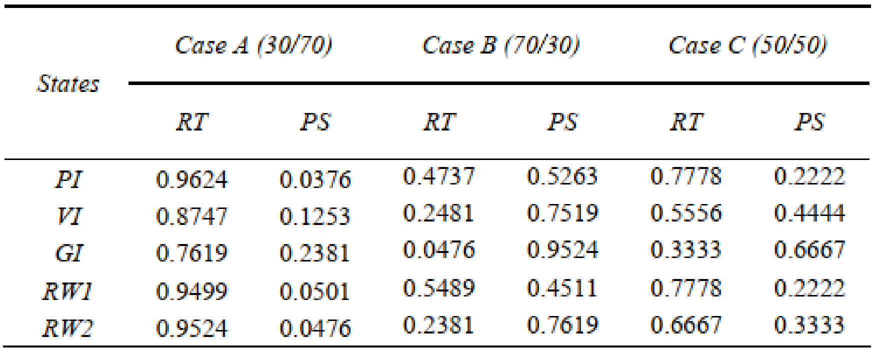
Moreover, the fundamental matrix can be used to calculate the expected number of transitions between states, which can be used to understand how often parts are inspected multiple times and how often parts are accepted or rejected on the first inspection. This information can be used to identify areas of the process where additional resources (e.g., more inspectors or more advanced inspection equipment) might be needed to reduce the number of inspections and speed up the process.

Overall, the fundamental matrix provides a powerful tool for analyzing the long-term behavior of the quality inspection process in the automotive industry and can be used to identify areas for improvement and make data-driven decisions about how to optimize the process.

4.2.6. RESULTS OF THE LIMITING MATRIX

The limiting matrix can be used to calculate the expected number of transitions between states, which can be used to understand how often parts are inspected multiple times and how often parts are accepted or rejected on the first inspection. The results for the three case A, B and C are presented below, in the Table 4.4

Table 4.4 Results of the limiting matrix for the three cases



States	*Case A (30/70)*		*Case B (70/30)*		*Case C (50/50)*	
	RT	*PS*	*RT*	*PS*	*RT*	*PS*
PI	0.9624	0.0376	0.4737	0.5263	0.7778	0.2222
VI	0.8747	0.1253	0.2481	0.7519	0.5556	0.4444
GI	0.7619	0.2381	0.0476	0.9524	0.3333	0.6667
RW1	0.9499	0.0501	0.5489	0.4511	0.7778	0.2222
RW2	0.9524	0.0476	0.2381	0.7619	0.6667	0.3333

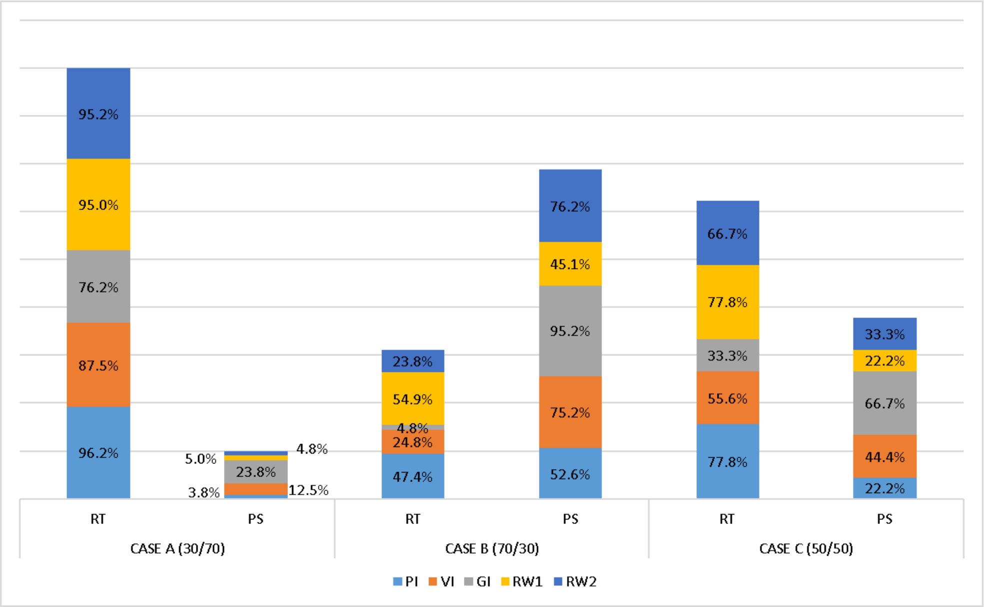


Figure 4. Limiting Matrix

4.2.6.1. DISCUSSION ON THE LIMITING MATRIX

In the automotive industry, one use of the limiting matrix is to calculate the probability of a part being accepted or rejected in the long-term, given that it starts in a particular state. This information can be used to understand the overall quality of the manufactured parts and identify areas of the process where the quality is lower than desired.

Additionally, the limiting matrix can be used to calculate the expected number of transitions between states, which can be used to understand how often parts are inspected multiple times and how often parts are accepted or rejected on the first inspection. This information can be used to identify areas of the process where additional resources (e.g., more inspectors or more advanced inspection equipment) might be needed to reduce the number of inspections and speed up the process.

Furthermore, the limiting matrix can be used to calculate the expected time spent in each state, which can be used to understand the efficiency of the inspection process and identify bottlenecks that are causing delays.

Overall, the limiting matrix provides a complementary tool to the fundamental matrix for analyzing the long-term behavior of the quality inspection process in the automotive industry, and can be used to identify areas for improvement and make data-driven decisions about how to optimize the process.

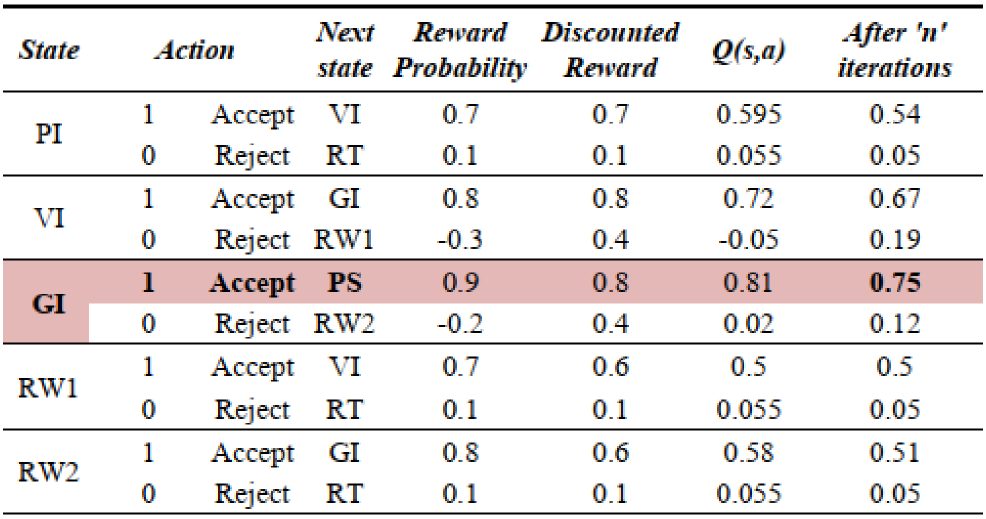
4.3 REINFORCEMENT LEARNING ALGORITHM

Reinforcement learning (RL) has been shown to be an effective tool for solving decision-making problems in operations, such as production planning and control, inventory management, and logistics.

4.3.1 DISCUSSION OF THE DYNAMIC PROGRAMMING RESULTS

This section contains information on the results of the Q-learning algorithm to estimate the optimal strategy to get the maximum reward. In the DP approach, we first compute value iteration to find the optimum value function. Further to that, a random policy is initialized and using the optimal value function. The iteration is carried until the optimum policy is reached. To achieve the optimum value function and the optimal policy through the DP approach, we need to have the model dynamics such as transition probability and reward probability. In the TD approach, both the optimum value function and optimal policy are estimated without any prior knowledge of the model dynamics. In both approaches, the learning rate was fixed at 0.3 for ease of explanation. In Table 1 presented in section 2.5, the reward values and discounted reward values are assumed arbitrarily.

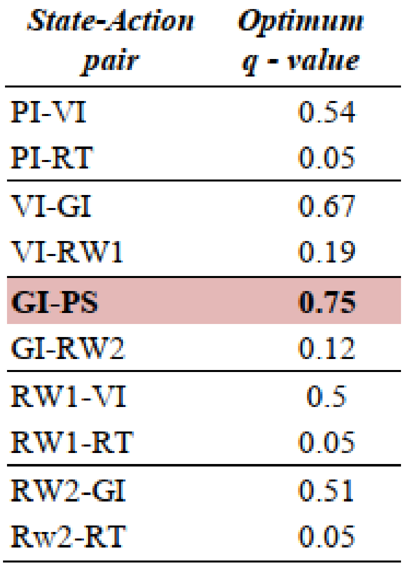
Table 4.3.1 Estimation of Q value using Dynamic Programming – value iteration.



State	*Action*		*Next state*	*Reward Probability*	*Discounted Reward*	*Q(s,a)*	*After 'n' iterations*
PI	1	Accept	VI	0.7	0.7	0.595	0.54
	0	Reject	RT	0.1	0.1	0.055	0.05
VI	1	Accept	GI	0.8	0.8	0.72	0.67
	0	Reject	RW1	-0.3	0.4	-0.05	0.19
GI	**1**	**Accept**	**PS**	0.9	0.8	0.81	**0.75**
	0	Reject	RW2	-0.2	0.4	0.02	0.12
RW1	1	Accept	VI	0.7	0.6	0.5	0.5
	0	Reject	RT	0.1	0.1	0.055	0.05
RW2	1	Accept	GI	0.8	0.6	0.58	0.51
	0	Reject	RT	0.1	0.1	0.055	0.05

Table 2 showcases the results of the q-values estimated using dynamic programming. After estimating the Q (s, a) with the model dynamics, the process is continued through a number of iterations to finally achieve the optimal value function. For the assumed reward and discounted reward values, the DP algorithm reached the optimum value function at the seven iterations. The optimality is assessed when there is no or less change between the iterated values function.

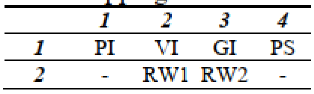
Table 4.x Estimation of optimum policy using policy iteration.



State-Action pair	*Optimum q - value*
PI-VI	0.54
PI-RT	0.05
VI-GI	0.67
VI-RW1	0.19
GI-PS	**0.75**
GI-RW2	0.12
RW1-VI	0.5
RW1-RT	0.05
RW2-GI	0.51
Rw2-RT	0.05

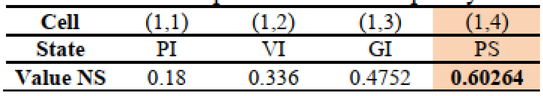
As detailed in section 2.4.3, policy iteration is a process of finding the value function for the optimum policy. Thereby after finding the value iteration, the optimum policy is estimated by step-by step evaluation of the randomly initialized policy. From the optimal value function, the maximum value (0.75) is chosen. This signifies that the agent taking action to move from GI to PS is the optimum policy. In the attempt to calculate the optimal policy for acceptance from the Start (Package Inspection) and to reach the destination (Pack and Store), Table 4 shows the cell mapping for the substations. Four possible policies are estimated to move from Start to Destination and are tabulated from Table 5 until Table 8.

Table 4. TD cell mapping table for Pack and Store



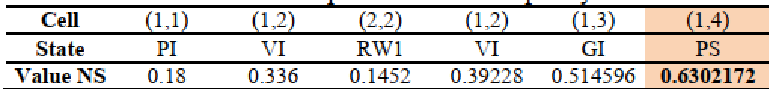
	1	*2*	*3*	*4*
1	PI	VI	GI	PS
2	-	RW1	RW2	-

Table 5. TD Update rule table – policy 1



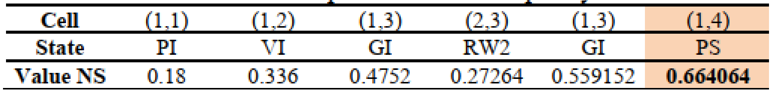
Cell	(1,1)	(1,2)	(1,3)	(1,4)
State	PI	VI	GI	PS
Value NS	0.18	0.336	0.4752	**0.60264**

Table 6. TD Update rule table – policy 2



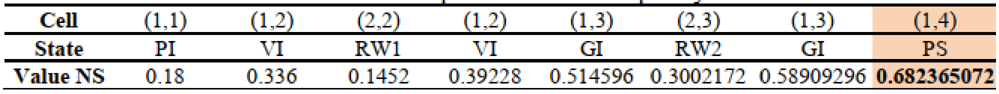
Cell	(1,1)	(1,2)	(2,2)	(1,2)	(1,3)	(1,4)
State	PI	VI	RW1	VI	GI	PS
Value NS	0.18	0.336	0.1452	0.39228	0.514596	**0.6302172**

Table 7. TD Update rule table – policy 3



Cell	(1,1)	(1,2)	(1,3)	(2,3)	(1,3)	(1,4)
State	PI	VI	GI	RW2	GI	PS
Value NS	0.18	0.336	0.4752	0.27264	0.559152	**0.664064**

Table 8. TD Update rule table – policy 4



Cell	(1,1)	(1,2)	(2,2)	(1,2)	(1,3)	(2,3)	(1,3)	(1,4)
State	PI	VI	RW1	VI	GI	RW2	GI	PS
Value NS	0.18	0.336	0.1452	0.39228	0.514596	0.3002172	0.58909296	**0.682365072**

Further, to calculate the optimal policy for rejection using the TD approach, Table 9 details the cell mappings from the Start (Packaging Inspection) and to reach the Goal (Return) substation. Table 10 until Table 13 display the values of the previous state.

Table 9. Temporal Distance cell mapping table for Return

	1	*2*	*3*
1	PI	VI	RW1
2	-	GI	-
3	-	RW2	RT

Table 10. TD Update rule table – policy 5

Cell	(1,1)	(3,3)
State	PI	RT
Value NS	0.18	**0.006**

Table 11. TD Update rule table – policy 6

Cell	(1,1)	(1,2)	(1,3)	(3,3)
State	PI	VI	RW1	RT
Value NS	0.18	0.336	0.1668	**0.00324**

Table 12. TD Update rule table – policy 7

Cell	(1,1)	(1,2)	(2,2)	(3,2)	(3,3)
State	PI	VI	GI	RW2	RT
Value NS	0.18	0.336	0.4752	0.30288	**0.092016**

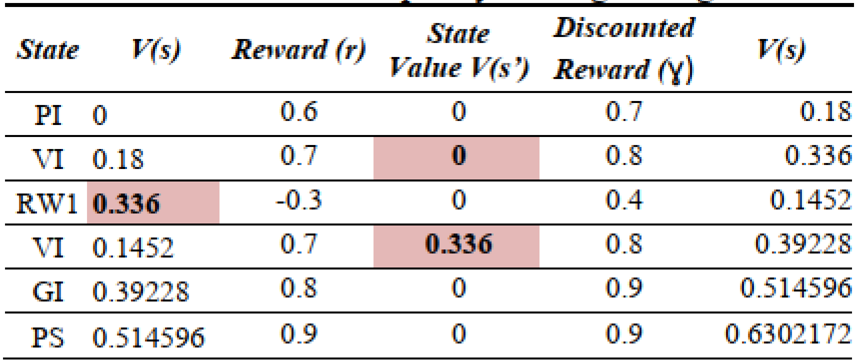
Table 13. TD Update rule table – policy 8

Cell	(1,1)	(1,2)	(1,3)	(1,2)	(2,3)	(3,2)	(3,3)
State	PI	VI	RW1	VI	GI	RW2	RT
Value NS	0.18	0.336	0.1668	0.35676	0.489732	0.3149208	**0.10044456**

Prediction of the outcome at the time (t+1) is better than the prediction at the time(t). Hence use the later prediction (t+1) to adjust the earlier prediction at the time (t). Prediction of outcome (judgment of the package reaching the Pack and Store) at gauge inspection (GI) is better than the prediction at package inspection (PI). Hence use the score or judgment at GI to adjust the prediction at PI. When a raw material tends to move from PI to VI, there is an equal probability (initial move, hence no experience is applied) that it would either qualify at station VI to go to GI or get disqualified and sent to RT. So, the intuition that the raw material would get qualified at VI (t+1) is better than the intuition at PI (t). Meaning, the reality or the fact about the quality of the raw material is more evident at each of the later stages than the initial stage.

To be more precise, consider Table 14 to explain the estimation of policy 2 displayed in Table 6. If we assume that the value of the previous state is '0' when the raw material enters VI from PI, but when the same raw materials re-enter station VI from Rework 1, the value of the state changes from '0' to '0.336'. As the value of state at (t+1) is improved, the value of state at (t) should also get improved. The results of the DP and TD algorithms arrive at the optimum policy.

Table 14. Estimation of policy 2 using TP algorithm.



State	*V(s)*	*Reward (r)*	*State Value V(s')*	*Discounted Reward (γ)*	*V(s)*
PI	0	0.6	0	0.7	0.18
VI	0.18	0.7	**0**	0.8	0.336
RW1	**0.336**	-0.3	0	0.4	0.1452
VI	0.1452	0.7	**0.336**	0.8	0.39228
GI	0.39228	0.8	0	0.9	0.514596
PS	0.514596	0.9	0	0.9	0.6302172

By comparing the results of the dynamic programming model and the Temporal Differencing models presented above, it is evident that the models produce similar results. However, the choice of the model depends on the availability of the model dynamics. DP algorithm is chosen when all the model parameters are known in advance, while TD algorithm is chosen otherwise when the model dynamics are unknown.

The above section 4.3.2 presented the Markov model developed to estimate the probability of the raw material being accepted or rejected in an incoming inspection environment. The proposed forecasting model is further optimized for efficiency using the two reinforcement learning algorithms (dynamic programming and temporal differencing). The TD models exploit the Markov property where the future states depend on the current state. This nature of the TD is more efficient in Markov environments. The results of the two optimized models are compared, and the findings are discussed.

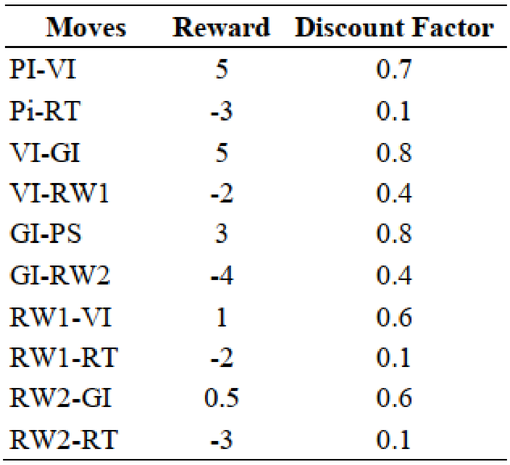
When the model dynamics are not known, and when we do not have any information about the environment and when there is a need to explore all possible policies, the go-to algorithm is the Monte Carlo approach. In MC, all the possible paths are attempted, and the search is extensive.

4.3.2 CLASSIFICATION OF MATERIAL QUALITY USING TD METHODS

This section demonstrates the use of Q-learning to estimate the possible next state. Every raw material that approaches the incoming inspection moves to all the substations for investigation. Depending on the acceptance and rejection at each stage, the trajectory of the material grows. The trajectory is used for grading the material for its quality.

During the estimation of the value of the state, the learning rate or step size is responsible for the convergence, and hence the learning rate (α) is chosen to be 0.1 for a smooth convergence based on the guidelines [3]. For every correct decision, the agent gets a reward, whereas, for every undesired action, the agent is penalized. The values for the rewards (r) and the penalization are represented as positive and negative integers, respectively. The discount factor (γ) is a concept that governs the relative relevance of immediate and future rewards. The discount factor and the reward and penalization for each move are tabulated below in Table 2. The choice of the reward and discount values were chosen arbitrarily.

Table 4.x Moves, rewards, and discount factors



Moves	Reward	Discount Factor
PI-VI	5	0.7
Pi-RT	-3	0.1
VI-GI	5	0.8
VI-RW1	-2	0.4
GI-PS	3	0.8
GI-RW2	-4	0.4
RW1-VI	1	0.6
RW1-RT	-2	0.1
RW2-GI	0.5	0.6
RW2-RT	-3	0.1

The numerical values that the agent obtains for completing some action at some state(s) in the environment are referred to as rewards. Based on the agent's activities, the numerical value might be positive or negative. In real life, we are more concerned with maximizing the cumulative reward (all the rewards the agent receives from the environment) than the reward the agent receives from the present condition (also called immediate reward). Returns refer to the overall amount of reward the agent receives from the environment.

The findings of the study are presented in this section. Q-learning algorithm for two situations: 1) Estimation of the quality index for the Pack and Store path and 2) quality index estimation for the return path. Table 3 demonstrates the four possible paths traveled by the accepted raw material. The substations are referenced as PI for Package

Inspection, VI for Visual Inspection, GI for Gauge Inspection, PS for Pack and Store, RW1 and RW2 for Rework 1 and Rework 2, respectively. Figure 3 depicts the temporal graph representation of the acceptance trajectory.

a) PI → VI → GI → PS

b) PI → VI → RWI → VI → GI → PS

c) PI → VI → GI → RW2 → GI → PS

d) PI → VI → RWI → VI → GI → RW2 → GI → PS

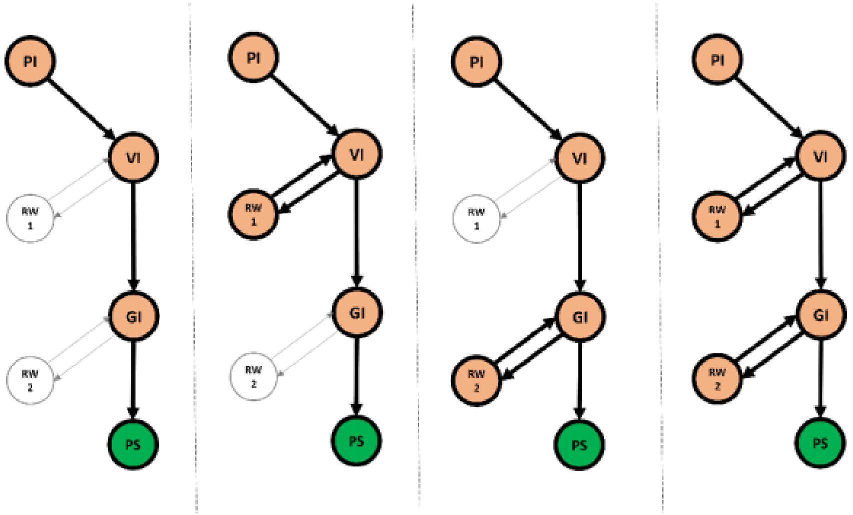
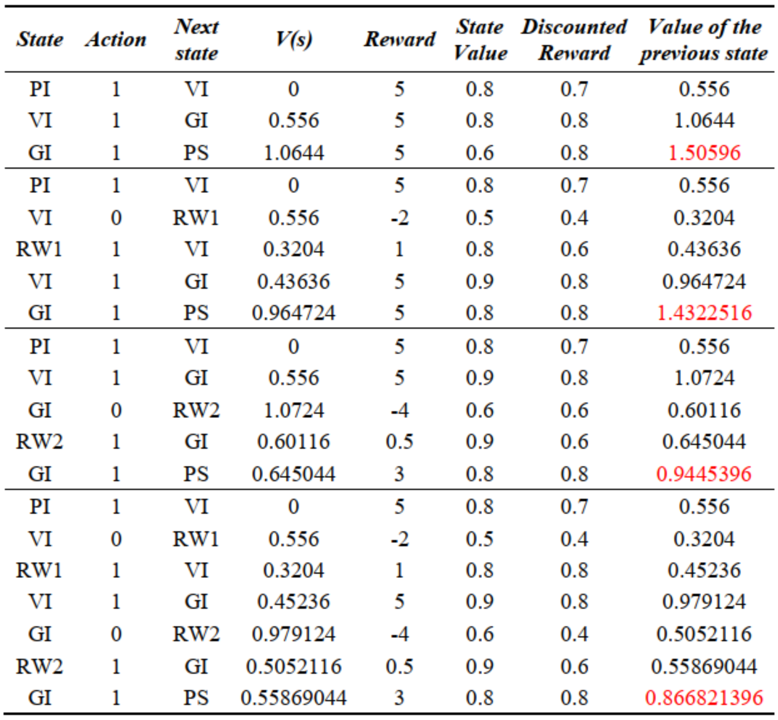


Figure 4.2 Temporal graph representation of the acceptance trajectory

Table 4.x Estimation of the quality index using TD for the Pack and Store path.



State	*Action*	*Next state*	*V(s)*	*Reward*	*State Value*	*Discounted Reward*	*Value of the previous state*
PI	1	VI	0	5	0.8	0.7	0.556
VI	1	GI	0.556	5	0.8	0.8	1.0644
GI	1	PS	1.0644	5	0.6	0.8	1.50596
PI	1	VI	0	5	0.8	0.7	0.556
VI	0	RW1	0.556	-2	0.5	0.4	0.3204
RW1	1	VI	0.3204	1	0.8	0.6	0.43636
VI	1	GI	0.43636	5	0.9	0.8	0.964724
GI	1	PS	0.964724	5	0.8	0.8	1.4322516
PI	1	VI	0	5	0.8	0.7	0.556
VI	1	GI	0.556	5	0.9	0.8	1.0724
GI	0	RW2	1.0724	-4	0.6	0.6	0.60116
RW2	1	GI	0.60116	0.5	0.9	0.6	0.645044
GI	1	PS	0.645044	3	0.8	0.8	0.9445396
PI	1	VI	0	5	0.8	0.7	0.556
VI	0	RW1	0.556	-2	0.5	0.4	0.3204
RW1	1	VI	0.3204	1	0.8	0.8	0.45236
VI	1	GI	0.45236	5	0.9	0.8	0.979124
GI	0	RW2	0.979124	-4	0.6	0.4	0.5052116
RW2	1	GI	0.5052116	0.5	0.9	0.6	0.55869044
GI	1	PS	0.55869044	3	0.8	0.8	0.866821396

Consider the results shown in Table 4.x above. The total score obtained for acceptance path A is 1.50596. This score is labelled for every raw material which follows this path. Similarly, for acceptance path B, the score is reduced to 1.4322516 as a penalty (negative reward) for undergoing a rework for visual correction. In acceptance path C, the score further reduces to 0.9445356 for a dimensional correction. The difference between acceptance paths B and C is due to the significance of the next state. Technically, the visual correction is less significant than dimension correction. When the raw material undergoes both the visual and dimensional correction, the score is now 0.866821396. The score for the raw materials traversing through the ideal path (path A) is 1.50596. The score for the raw materials following path B is 7.37% lower than the ideal path. Path C and D are 37.28% and 42.44%, respectively lower than the ideal path. Therefore, any irregularities in the dimension pave the way for more penalization. Every raw material now gets a label after the incoming inspection process signifying the trajectory and thus the quality of the accepted material. Figure 4 showcases the temporal graph representations of the rejection trajectory.

Table 3 lists the four possible paths traversed by the raw materials to reach the Return (RT).

e) PI → RT

f) PI → VI → RWI → RT

g) PI → VI → GI → RW2 → RT

h) PI → VI → RWI → VI → GI → RW2 → RT

Fig 4.3 Temporal graph representation of the rejection trajectory

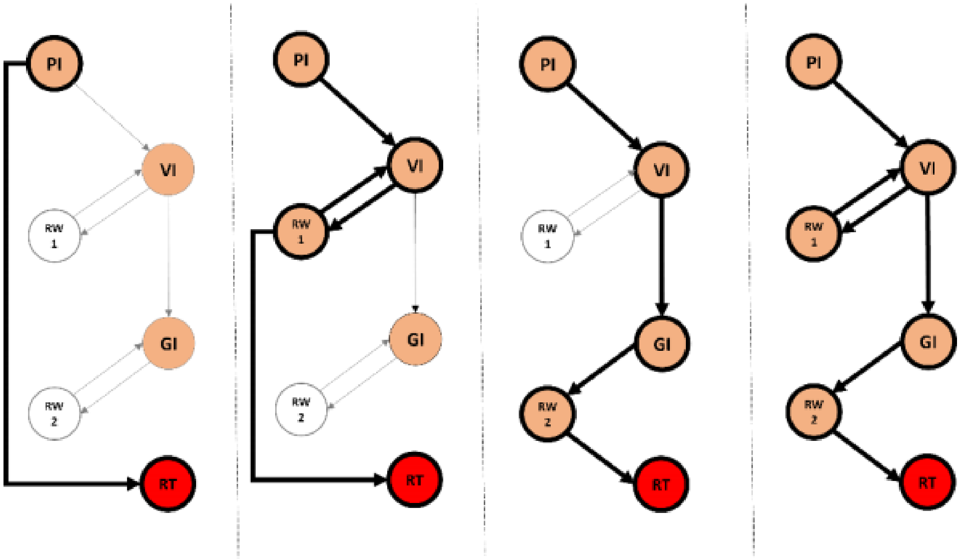


Table 4.x Estimation of the quality index using TD for Return path.

State	***Action***	***Next state***	***V(s)***	***Learning rate***	***Reward***	***State Value***	***Discounted Reward***	***Value of the previous state***
PI → RT								
PI	1	RT	0	0.1	-3	0.1	0.1	-0.299
PI → VI → RWI → RT								
PI	1	VI	0	0.1	5	0.8	0.7	0.556
VI	0	RW1	0.556	0.1	-2	0.5	0.4	0.3204
RW1	0	RT	0.3204	0.1	-2	0.1	0.1	0.08936
PI → VI → GI → RW2 → GI → RT								
PI	1	VI	0	0.1	5	0.8	0.7	0.556
VI	1	GI	0.556	0.1	5	0.9	0.8	1.0724
GI	0	RW2	1.0724	0.1	-4	0.6	0.4	0.58916
RW2	1	RT	0.58916	0.1	-3	0.1	0.1	0.231244
PI → VI → RWI → VI → GI → RW2 → GI → RT								
PI	1	VI	0	0.1	5	0.8	0.7	0.556
VI	0	RW1	0.556	0.1	-2	0.5	0.4	0.3204
RW1	1	VI	0.3204	0.1	1	0.8	0.6	0.43636
VI	1	GI	0.43636	0.1	5	0.9	0.8	0.964724
GI	0	RW2	0.964724	0.1	-4	0.6	0.4	0.4922516
RW2	1	RT	0.4922516	0.1	-3	0.1	0.1	0.14402644

The results for the rejection path are shown in Table 4. Packaging inspection has an essential role to play in the incoming inspection process. Irrespective of the material quality, the package is judged for adherence to the packaging standards. Upon failure, the raw materials are directly rejected and sent to the supplier with a total score of -0.299. In path F, the subsequent rejection stage can occur at Rework station 1 (RW1) due to visual nonconformities with a total rejection score of 0.08936. Fit and function inspection using a gauge plays a vital role in checking dimensional conformity. The raw materials failing at this stage get the lower total score of 0.231244. If a raw material fails in both visual and gauge inspection is awarded the least total score of 0.14402644. The supplier usually scraps these materials.

The above section, an incoming inspection problem is considered as a reinforcement learning task. A Temporal difference learning approach predicts the acceptance and rejection path of raw materials in the incoming inspection process. The algorithm presented eight possible paths that the raw materials could travel. Four trajectories contribute to material acceptance, whereas the remaining paths lead to material rejection. The materials are labelled using the total scores obtained in the incoming inspection process. The materials traveling on the ideal path (path A) get the highest total score. The rest of the accepted materials have a 7.37% lower score in path B, whereas path C and path D get 37.28% and 42.44% lower from the ideal path.

4.4 DEEP ACTIVE LEARNING AND COMPUTER VISION BASED VISUAL INSPECTION

The use of DAL and computer vision-based visual inspection can lead to significant improvements in the accuracy and efficiency of visual inspection tasks. DAL can help to reduce the number of samples that need to be annotated, which can save time and money.

4.4.1 DEEP ACTIVE LEARNING RESULTS FOR CASTING DATASET

This section demonstrates the results of the deep active learning technique applied for the image classification of the casting dataset. The following are the sampling algorithms used in this research work.

1. Entropy algorithm results
2. Random algorithm results
3. Deep Bayesian Active Learning (DBAL) algorithm results
4. Bayesian Active Learning by Disagreement (BALD) algorithm results

4.4.1.1 ENTROPY SAMPLING RESULTS

This section presents the results for the resented in Figure 4.3 shows the impact of applying entropy sampling on the casting dataset.

In Table 4.3 a more detailed view of the ten first rounds and their accuracy are demonstrated.

Table 4.x Results of the Entropy query sampling algorithm

Episodes	**PL**	**AL1**	**AL2**	**AL3**	**AL4**	**AL5**
% Of Labelled samples	10	14.5	19	23.6	32.6	37.1
Entropy	96.92	98.88	**99.72**	**99.86**	**99.86**	**99.72**

The highest accuracy achieved is 98.58% with a learning rate of 0.05 and with 8500 labelled samples. In Section 4.1.6 we will extend the results by evaluating the model for more labelled samples with a learning rate set to 0.05.

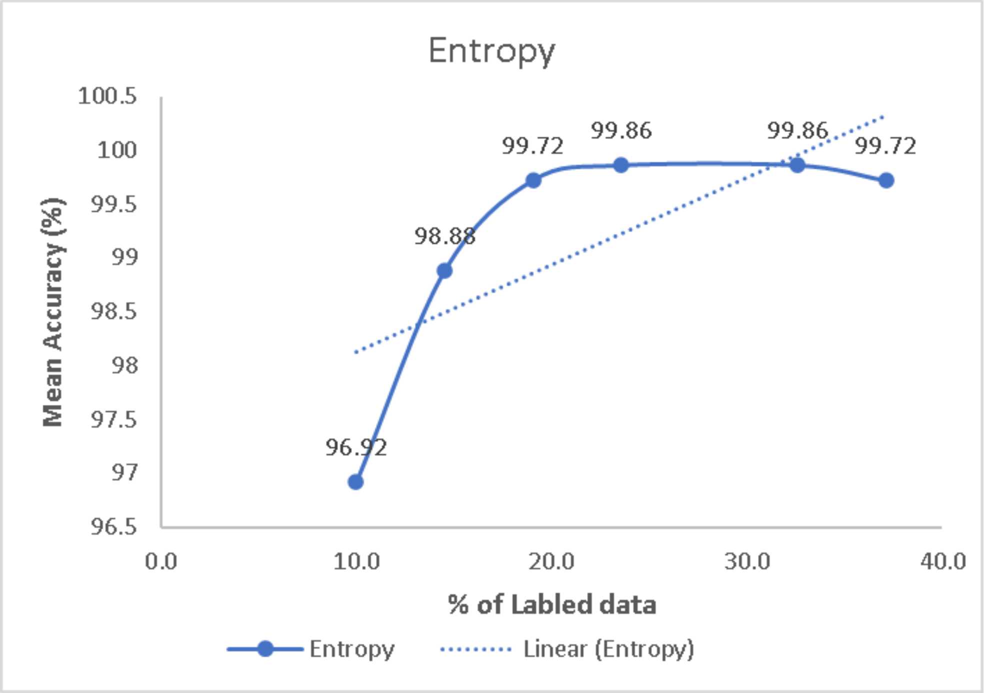


Figure 4.x: Effect of learning rate scheduler using entropy query sampling.

4.4.1.2 RANDOM SAMPLING RESULTS

The results presented in Figure 4.3 shows the impact of applying random sampling on the casting dataset.

In Table 4.3 a more detailed view of the ten first rounds and their accuracy are demonstrated.

Table 4.x Results of the random query sampling algorithm

Episodes	**PL**	**AL1**	**AL2**	**AL3**	**AL4**	**AL5**
% Of Labelled samples	10	14.5	19	23.6	32.6	37.1
Random	96.64	97.62	95.66	98.46	98.6	99.58

The highest accuracy achieved is 98.58% with a learning rate of 0.05 and with 8500 labelled samples. In Section 4.1.6 we will extend the results by evaluating the model for more labelled samples with a learning rate set to 0.05.

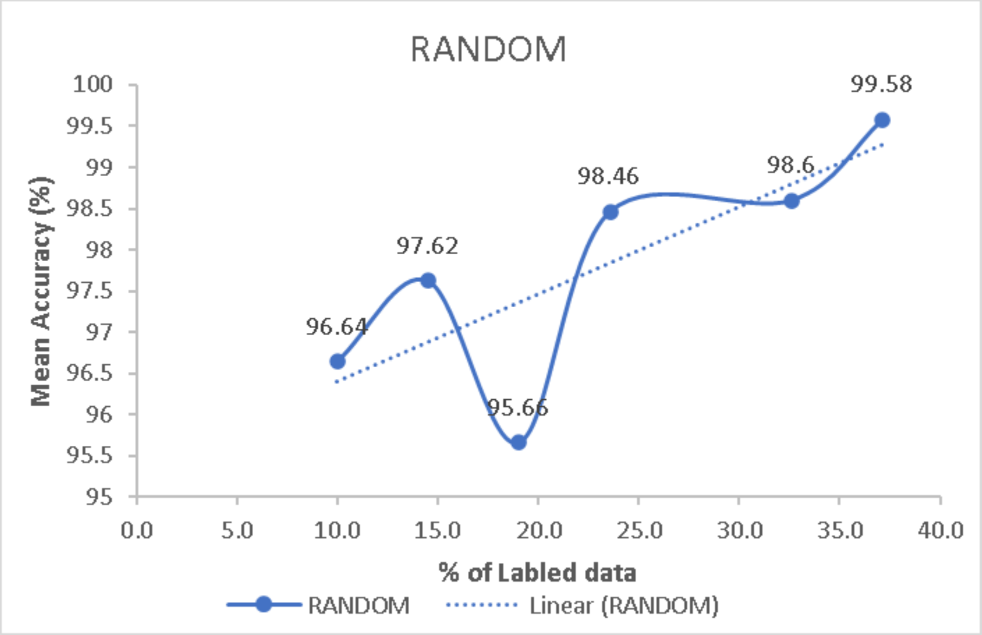


Figure 4.x: Effect of learning rate scheduler using random query sampling.

4.4.1.3 DBAL SAMPLING RESULTS

The results presented in Figure 4.3 shows the impact of applying DBAL sampling on the casting dataset.

In Table 4.3 a more detailed view of the ten first rounds and their accuracy are demonstrated.

Table 4.x Results of the DBAL query sampling algorithm

Episodes	**PL**	**AL1**	**AL2**	**AL3**	**AL4**	**AL5**
% Of Labelled samples	10	14.5	19	23.6	32.6	37.1
DBAL	97.6	98.88	98.88	99.44	99.44	99.58

The highest accuracy achieved is 98.58% with a learning rate of 0.05 and with 8500 labelled samples. In Section 4.1.6 we will extend the results by evaluating the model for more labelled samples with a learning rate set to 0.05.

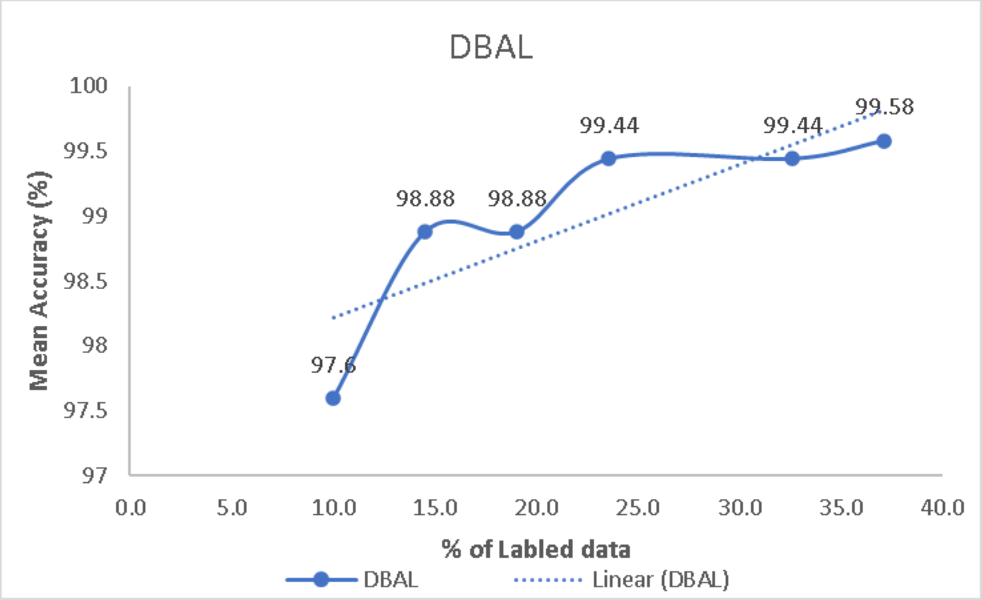


Figure 4.x: Effect of learning rate scheduler using DBAL query sampling.

4.4.1.4 BALD SAMPLING RESULTS

The results for the BALD query sampling algorithm applied on the casting dataset is presented in Table 4.x. Figure 4.3 shows the performance of the BALD query sampling algorithm under the learning rate scheduler on the casting dataset.

Table 4.x Results of the BALD query sampling algorithm

Episodes	**PL**	**AL1**	**AL2**	**AL3**	**AL4**	**AL5**
% Of Labelled samples	10	14.5	19	23.6	32.6	37.1
BALD	96.08	**99.02**	99.58	99.72	99.44	**99.72**

The highest accuracy achieved is 99.02% with 14.5% of the labelled samples.

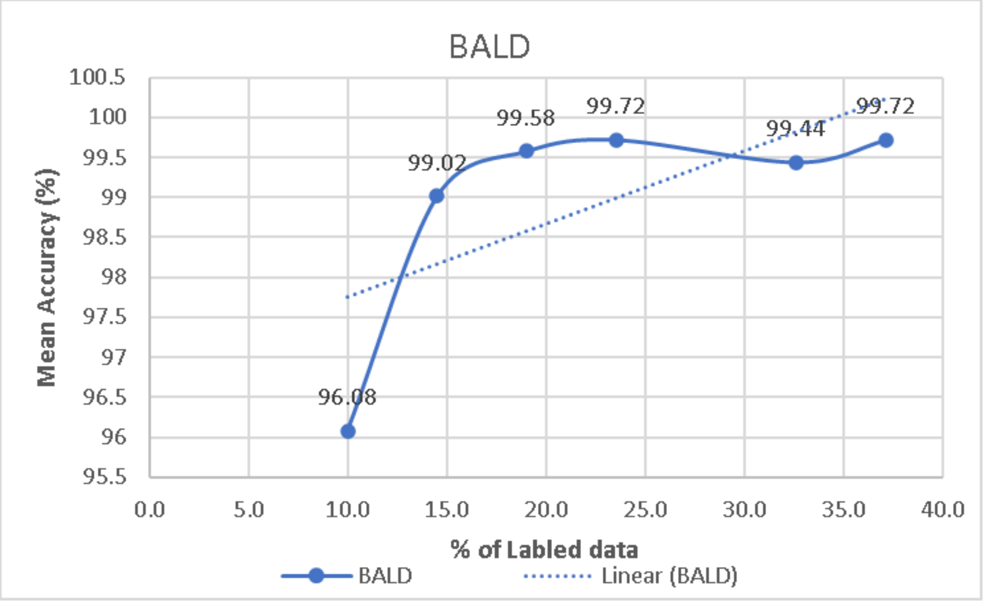


Figure 4.x: Effect of learning rate scheduler using BALD query sampling.

4.4.1.5 COMPARISON OF QUERY SAMPLING METHODS

This section consolidates the results of the query sampling methods under the learning rate scheduler. The results of the Resnet model trained on the casting dataset under the query sampling algorithms are tabulated in Table 4.x below. The base model is trained for 1 episode using the traditional passive learning (PL) approach with 10% of labelled dataset. Subsequently, the deep active learning approach is applied by picking up approximately 4.5% of unlabelled samples from the total dataset. The percentage of labelled samples for each episode is tabulated in Table 4.x. The active learning runs for 5 iterations and the result for each episode is tabulated in Table 4.x.

Table 4.x Results of the Resnet model under various query sampling algorithms

Episodes	**PL**	**AL1**	**AL2**	**AL3**	**AL4**	**AL5**
% Of Labelled samples	10	14.5	19	23.6	32.6	37.1
Number of labelled samples	663	693	1263	1563	2163	2463

Table 4.x Results of the Resnet model under various query sampling algorithms

Episodes	**PL (%)**	**AL1 (%)**	**AL2 (%)**	**AL3 (%)**	**AL4 (%)**	**AL5 (%)**
Entropy	96.92	98.88	99.72	**99.86**	99.86	99.72
Random	96.64	97.62	95.66	98.46	98.6	**99.58**
DBAL	97.60	98.88	98.88	99.44	99.44	**99.58**
BALD	96.08	99.02	99.58	**99.72**	99.44	99.72

Figure 4.x depicts the performance comparison of the query sampling algorithms on the casting dataset. The graph shows the model accuracy against the number of labelled samples used for the passive learning and deep active learning approaches.

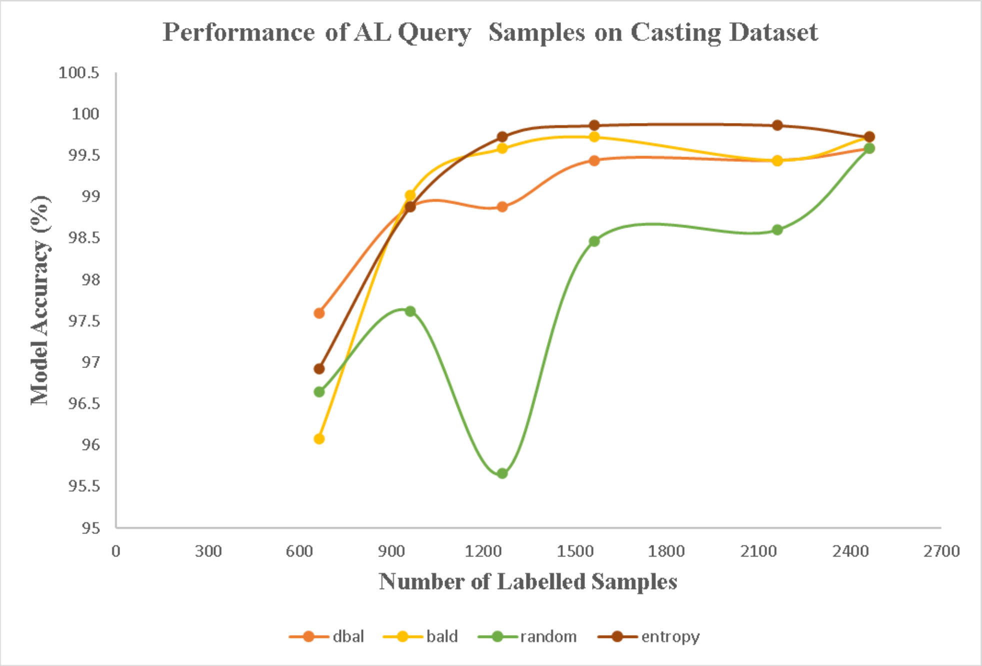


Figure 4.x: Performance comparison of deep active learning query sampling algorithms on the casting dataset

Table 4.x Comparison of performance against the number of labelled samples

Authors	Number of labelled samples for training	Model Architecture	Model Accuracy (%)
Nguyen et al (H. T. Nguyen et al., 2020)	5993	Resnet	93.01
	5993	Densenet	95.94
	5993	Googlenet	95.24
Nguyen et al (Huy Toan et al., 2021a)	5144	VGGnet	97.76
	5144	Resnet	98.46
	5144	Densenet	99.58
	5144	Googlenet	99.30
Kim et al (Kim et al., 2022a)	5000	Custom CNN	98.46
Proposed Deep Active Learning Approach	1563	Resnet18-entropy	99.86
	2463	Resnet18-random	99.58
	2463	Resnet18-DBAL	99.58
	1563	Resnet18-BALD	99.72

Table 4.x presents the comparison of the number of labelled training samples used for model training on casting dataset.

4.4.2 DISCUSSION ON THE RESULTS FOR DEEP ACTIVE LEARNING FOR CASTING DATASET

The results for the classification models trained using the deep active learning approach are tabulated in section 4.4.1. The overall performance of the classification models outshines the traditional deep learning (passive) models.

Nguyen proposed a Deep Learning approach to classify the defective products using the convolutional neural network (Huy Toan et al., 2021b). In their work, they have used all the images 6653 from the dataset.

Kim and his associates developed a CNN model to classify the casting dataset based on the augmented dataset with the image dimensions 512 x 512. (Kim et al., 2022b)

Their training is based on the conventional passive learning approach. The model training set involved labelled dataset and reported 98.46% accuracy with 5000 samples.

The base model also referred to as the passive learning model is built from 10% of the labelled samples which is 663 samples from the overall dataset. This set of 663 labelled samples are chosen in random during the deep learning model training.

After the base model is successfully trained, the active learning training starts by adding a certain number of samples per each iteration which is referred to as the budget size. In this scenario, the budget size is set to 300 or 4.5%. This budget size depends on the training dataset and the number of active learning iterations.

The active learning training started with 14.5% samples which is 943 samples in the first iteration. At each iteration thereof, another 300 samples or approximately 4.5% is added to the labelled pool of dataset.

The query sampling algorithms are responsible for how these 300 samples are chosen from the pool of unlabelled samples. In this experiment, the four query sampling algorithms are chosen based on its performance as presented in the literature (Kumar & Gupta, 2020; Ren et al., 2022; Settles, 2009; Zhan, Wang, et al., 2022c).

From the comparison table shown in table 4.x, it is evident that active learning outperforms the model performance with 1/4 of the total dataset. While comparing among the query sampling algorithms, the entropy sampling technique performs better and takes lesser training time compared to BALD sampling algorithm.

4.5 SUMMARY

Chapter 4 showed the optimization model findings. This chapter summarized and compared the results to prior studies. Sections 1–3 make up the chapter. Section 4.2 presents the Markov decision process-based raw material acceptance prediction model for the arrival inspection case study. The Markov properties and related studies are then applied to the findings. Section 4.3 presents the optimized models for raw material acceptance. These models employ dynamic programming and temporal difference. After comparing results, a detailed discussion follows. The material quality model's results are also presented and analysed. Section 4.4 presents deep learning computer vision model-assisted visual inspection results. Traditional and deep active learning outcomes have been expanded on and analysed. The following final chapter of this research work encapsulates the overall findings and relates it back to the research objectives and the recommendations for future research are highlighted.

CHAPTER 5

CHAPTER FIVE: CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

This thesis developed three components to address the objectives highlighted in the initial chapters. The research methodology is divided up into three main components, Firstly, the Markov decision technique and an inbound inspection case study that makes use of RFID tag data was considered.. A state machine diagram has been built so that the results of the case study may be understood more clearly. Full explanations are given for both the case study for the inbound inspection and the substations that had a role in it. Further, a more in-depth demonstration of the computation of Markov chain characteristics for the experimental situations is provided. In addition, the transition matrix as well as an analysis of the transition probability as well as the procedures that were utilised for the creation of the raw material acceptance prediction model are described. All of these things were done in order to create the raw material acceptance prediction model.

Secondly, the prediction model that was built using the Markov decision process that was solved in the previous sections is optimised through the use of reinforcement learning, which is detailed in depth in that part. This is done to make the model as accurate as possible. This part then goes on to detail the steps that need to be followed to solve the raw material acceptance prediction model by utilising the temporal difference RL algorithm and the dynamic programming approach. These are the topics that are covered in the next section. In a similar manner, this part outlines the procedure that has to be completed in order to develop a model for the estimation of the material quality categorization utilising the temporal difference approach.

Thirdly, a comprehensive description of the impeller defects casting dataset is provided, along with an overview of the usual deep learning approach to visual inspection. This part goes into further depth regarding the processes that need to be done in order to maximise the model development by making the best use of the data based on a deep active learning approach. The goal of this section is to maximise the efficiency with which the model is developed.

5.2 RECOMMENDATIONS

Further to the current research, the following suggestions for the betterment of the quality 4.0.

1. Use of Markov Logic Networks
2. Exploration of computer vision applications to monitor the human behaviour for the better quality of life.

REFERENCES

Abhishek, N., & Biswas, M. (2018). Reinforcement learning with Open AI, TensorFlow and Keras Using Python. In *Apress* (Vol. 3, Issue 9). https://www.pdfdrive.com/reinforcement-learning-with-open-ai-tensorflow-and-keras-using-python-e158327149.html

Adams, R. P., & Elements, C. O. S. (n.d.). *Reinforcement Learning*. 1–14.

Adrian, L., Mayer, C., & Timofte, R. (2021). *Best Practices in Pool Based Active Learning for Image Classification*. https://github.com/Mephisto405/Learning-Loss-for-Active-Learning

Agarwal, U. (2022). *Active and Incremental Deep learning with class imbalanced data*.

Agerskans, N. (2019). *A Framework for Achieving Data-Driven Decision Making in Production Development*.

Aggarwal, U., Popescu, A., & Hudelot, C. (2022). *Optimizing Active Learning for Low Annotation Budgets*. http://arxiv.org/abs/2201.07200

Akanksha, E., Jyoti, Sharma, N., & Gulati, K. (2021). Review on Reinforcement Learning, Research Evolution and Scope of Application. *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, *May*, 1416–1423. https://doi.org/10.1109/ICCMC51019.2021.9418283

al Kattan, I., & AI-Khudairi, T. (2007). Improving supply chain management effectiveness using RFID. *IEEE International Engineering Management Conference*, *May*, 191–198. https://doi.org/10.1109/IEMC.2007.5235073

Anahideh, H., Asudeh, A., & Thirumuruganathan, S. (2022). Fair active learning. In *Expert Systems with Applications* (Vol. 199, Issue 1). Association for Computing Machinery. https://doi.org/10.1016/j.eswa.2022.116981

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). A Brief Survey of Deep Reinforcement Learning. *IEEE Signal Processing Magazine*, *34*(6), 26–38. https://doi.org/10.1109/msp.2017.2743240

Atighehchian, P., Branchaud-Charron, F., & Lacoste, A. (2020). *Bayesian active learning for production, a systematic study and a reusable library*. http://arxiv.org/abs/2006.09916

Aurobindo Munagala, S., Subramanian, S., Karthik, S., Prabhu, A., Namboodiri, A., & Hyderabad, I. (n.d.). *CLACTIVE: EPISODIC MEMORIES FOR RAPID ACTIVE LEARNING*.

Azcue, P., & Muler, N. (2014). Stochastic Optimization in Insurance: A Dynamic Programming Approach. In *SpringerBriefs in Quantitative Finance*. http://www.amazon.com/Stochastic-Optimization-Insurance-SpringerBriefs-Quantitative/dp/1493909940/ref=sr_1_1_title_0_main?s=books&ie=UTF8&qid=1404838238&sr=1-1

Baines, T., Mason, S., Siebers, P. O., & Ladbrook, J. (2004). Humans: the missing link in manufacturing simulation? *Simulation Modelling Practice and Theory*, *12*(7–8), 515–526. https://doi.org/10.1016/S1569-190X(03)00094-7

Barde, S. R. A., Yacout, S., & Shin, H. (2019). Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks. *Journal of Intelligent Manufacturing*, *30*(1), 147–161. https://doi.org/10.1007/s10845-016-1237-7

Barto, A., Thomas, P., & Sutton, S. R. (2017). Some recent applications of reinforcement learning. *In Proceedings of the Eighteenth Yale Workshop on Adaptive and Learning Systems.* https://people.cs.umass.edu/~pthomas/papers/Barto2017.pdf

Batson, R. G., & McGough, K. D. (2007). A new direction in quality engineering: Supply chain quality modelling. *International Journal of Production Research*, *45*(23), 5455–5464. https://doi.org/10.1080/00207540701325140

Beck, N., Sivasubramanian, D., Dani, A., Ramakrishnan, G., & Iyer, R. (2021a). *Effective Evaluation of Deep Active Learning on Image Classification Tasks*. 1–24. http://arxiv.org/abs/2106.15324

Beck, N., Sivasubramanian, D., Dani, A., Ramakrishnan, G., & Iyer, R. (2021b). *Effective Evaluation of Deep Active Learning on Image Classification Tasks*. http://arxiv.org/abs/2106.15324

Beluch, W. H., Genewein, T., Nürnberger, A., & Köhler, J. M. (2018). The Power of Ensembles for Active Learning in Image Classification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2018.00976

Belusso, C. L. M., Sawicki, S., Roos-Frantz, F., & Frantz, R. Z. (2016). A Study of Petri Nets, Markov Chains and Queueing Theory as Mathematical Modelling Languages Aiming at the Simulation of Enterprise Application Integration Solutions: A First Step. *Procedia Computer Science*, *100*, 229–236. https://doi.org/10.1016/j.procs.2016.09.147

Bengar, J. Z., Raducanu, B., & van de Weijer, J. (2021). When Deep Learners Change Their Mind: Learning Dynamics for Active Learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13052 LNCS*, 403–413. https://doi.org/10.1007/978-3-030-89128-2_39

Bertsekas, D. P. (2010). Pathologies of temporal difference methods in approximate dynamic programming. *Proceedings of the IEEE Conference on Decision and Control*, 3034–3039. https://doi.org/10.1109/CDC.2010.5717644

Bertsekas, D. P. (2012). Dynamic Programming and Optimal Control (2 Vol Set). In *Athena Scientific* (4th ed.). Athena Scientific.

Bone, C., & Dragićević, S. (2010). Simulation and validation of a reinforcement learning agent-based model for multi-stakeholder forest management. *Computers, Environment and Urban Systems*, *34*(2), 162–174. https://doi.org/10.1016/j.compenvurbsys.2009.10.001

Brachman, R. J., Research, Y. !, Domingos, P., & Lowd, D. (2009). *Markov Logic: An Interface Layer for Artificial Intelligence*.

Buffett, S. (2005). A Markov model for inventory level optimization in supply-chain management. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3501 LNAI*, 133–144. https://doi.org/10.1007/11424918_15

Caramalau, R., Bhattarai, B., Stoyanov, D., & Kim, T.-K. (2023). *MoBYv2AL: Self-supervised Active Learning for Image Classification*. 1–13. http://arxiv.org/abs/2301.01531

Chandra, A. L., & Balasubramanian, V. N. (2021). Deep Active Learning Toolkit for Image Classification in PyTorch. *Https://Github.Com/Acl21/Deep-Active-Learning-Pytorch*.

Chen, J., Schein, A., Ungar, L., & Palmer, M. (2006). *An Empirical Study of the Behavior of Active Learning for Word Sense Disambiguation*. http://www.senseval.org

Chen, M., Wang, T., Zhang, S., & Liu, A. (2021). Deep reinforcement learning for computation offloading in mobile edge computing environment. *Computer Communications*, *175*, 1–12. https://doi.org/10.1016/j.comcom.2021.04.028

Chen, S., Wang, T., & Jia, R. (2021). *Zero-Round Active Learning*. http://arxiv.org/abs/2107.06703

Contardo, G., Denoyer, L., Artières, T., Thierry, A., & Artieres, T. (2017). *A Meta-Learning Approach to One-Step Active-Learning*. https://hal.science/hal-01691472

Deisenroth, M. P. (2011). A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, *2*(1–2), 1–142. https://doi.org/10.1561/2300000021

Desai, S. V., Chandra, A. L., Guo, W., Ninomiya, S., & Balasubramanian, V. N. (2019). *An Adaptive Supervision Framework for Active Learning in Object Detection*. http://arxiv.org/abs/1908.02454

Desai, S. V., Lagandula, A. C., Guo, W., Ninomiya, S., & Balasubramanian, V. N. (2020). An adaptive supervision framework for active learning in object detection. *30th British Machine Vision Conference 2019, BMVC 2019*, 1–13.

Dhabi, R. (2019, October 22). *casting product image data for quality inspection | Kaggle*. Pilot Techno Cast. https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product

Doltsinis, S., Ferreira, P., & Lohse, N. (2014). An MDP model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*(9), 1125–1138. https://doi.org/10.1109/TSMC.2013.2294155

Drury, C. G., & Watson, J. (2002). *HUMAN FACTORS GOOD PRACTICES IN VISUAL INSPECTION*.

Elmachtoub, A. N., Liang, J. C. N., & McNellis, R. (2020). Decision trees for decision-making under the predict-then-optimize framework. *37th International Conference on Machine Learning, ICML 2020*, *PartF16814*, 2838–2847.

Fang, M., Li, Y., & Cohn, T. (2017a). Learning how to active learn: A deep reinforcement learning approach. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 595–605. https://doi.org/10.18653/v1/d17-1063

Fang, M., Li, Y., & Cohn, T. (2017b). *Learning how to Active Learn: A Deep Reinforcement Learning Approach*. http://arxiv.org/abs/1708.02383

Fenjiro, Y., & Benbrahim, H. (2018). Deep Reinforcement Learning Overview of the state of the Art. *Journal of Automation, Mobile Robotics & Intelligent Systems*, *12*(3). https://doi.org/10.14313/JAMRIS_3-2018/15

Forero, D. V., & Sisodia, R. (2020). *Quality 4 . 0 – How to Handle Quality in the Industry 4 . 0 Revolution Master ' s thesis in Quality and Operations Management*. *January*, 1–64. https://odr.chalmers.se/bitstream/20.500.12380/300650/1/E2019_128.pdf

Gal, Y., Islam, R., & Ghahramani, Z. (2017). Deep Bayesian active learning with image data. *34th International Conference on Machine Learning, ICML 2017*, *3*, 1923–1932.

Geifman, Y., & El-Yaniv, R. (n.d.). *Deep Active Learning with a Neural Architecture Search*.

Geifman, Y., & El-Yaniv, R. (2017). *Deep Active Learning over the Long Tail*. *m*, 1–10. http://arxiv.org/abs/1711.00941

Gingu, E.-I., & Zapciu, M. (2017). MARKOV CHAINS AND DECOMPOSITION METHOD USED FOR SYNCHRONIZING THE MANUFACTURING PRODUCTION RATE WITH REAL MARKET DEMAND. *U.P.B. Sci. Bull., Series D*, *79*.

Goldsby, T. J., & Martichenko, Robert. (2014). *Lean Six Sigma Logistics*. 301. https://www.perlego.com/book/1355074/lean-six-sigma-logistics-strategic-development-to-operational-success-pdf

Gosavi, A. (2019). *A Tutorial for Reinforcement Learning*. http://simoptim.comCodes:http://simoptim.com/bookcodes.html

Gronauer, S., & Diepold, K. (2022). Multi-agent deep reinforcement learning: a survey. In *Artificial Intelligence Review* (Vol. 55, Issue 2). Springer Netherlands. https://doi.org/10.1007/s10462-021-09996-w

Gruszka, J., & Misztal, A. (2017). The new IATF 16949:2016 standard in the automotive supply chain. *Research in Logistics and Production*, *7*(4), 311–318. https://doi.org/10.21008/j.2083-4950.2017.7.4.3

Gudovskiy, D., Hodgkinson, A., Yamaguchi, T., & Tsukizawa, S. (2020). Deep active learning for biased datasets via fisher kernel self-supervision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 9038–9046. https://doi.org/10.1109/CVPR42600.2020.00906

Habib, N. (2019). *Hands-on Q-learning with Python : practical Q-learning with OpenAI Gym, Keras, and TensorFlow*. Packt Publishing Ltd.

Haussmann, E., Ivanecky, J., Fenzi, M., Alvarez, J., Chitta, K., Roy, D., & Mittel Akshita. (2019). *Scalable Active Learning for Autonomous Driving*. Medium. https://medium.com/nvidia-ai/scalable-active-learning-for-autonomous-driving-a-practical-implementation-and-a-b-test-4d315ed04b5f

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. http://arxiv.org/abs/1512.03385

Hemmer, P., Kühl, N., & Schöffer, J. (n.d.). *DEAL: Deep Evidential Active Learning for Image Classification*.

Hemmer, P., Kühl, N., & Schöffer, J. (2020). *DEAL: Deep Evidential Active Learning for Image Classification*. http://arxiv.org/abs/2007.11344

Hu, H., Wu, Q., Zhang, Z., & Han, S. (2019). Effect of the manufacturer quality inspection policy on the supply chain decision-making and profits. *Advances in Production Engineering And Management*, *14*(4), 472–482. https://doi.org/10.14743/apem2019.4.342

Huang, K.-H. (2021a). *DeepAL: Deep Active Learning in Python*. *0*, 6–9. http://arxiv.org/abs/2111.15258

Huang, K.-H. (2021b). *DeepAL: Deep Active Learning in Python*. http://arxiv.org/abs/2111.15258

Huang, S. H., & Pan, Y. C. (2015). Automated visual inspection in the semiconductor industry: A survey. In *Computers in Industry* (Vol. 66). https://doi.org/10.1016/j.compind.2014.10.006

Huang, W., Sun, S., Lin, X., Zhang, D., & Ma, L. (2022). Deep active learning with Weighting filter for object detection. *Displays*, *76*(August 2022), 102282. https://doi.org/10.1016/j.displa.2022.102282

Hulbert, N., Spillers, S., Francis, B., Haines-Temons, J., Romero, K. G., de Jager, B., Wong, S., Flora, K., Huang, B., & Irissappane, A. A. (2021). EasyRL: A Simple and Extensible Reinforcement Learning Framework. *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, *18*, 16041–16043. https://doi.org/10.1609/aaai.v35i18.18006

Huy Toan, N., Yu, G. H., Shin, N. R., Kwon, G. J., Kwak, W. Y., & Kim, J. Y. (2021a). Defective product classification system for smart factory based on deep learning. *Electronics (Switzerland)*, *10*(7). https://doi.org/10.3390/electronics10070826

Huy Toan, N., Yu, G. H., Shin, N. R., Kwon, G. J., Kwak, W. Y., & Kim, J. Y. (2021b). Defective product classification system for smart factory based on deep learning. *Electronics (Switzerland)*, *10*(7). https://doi.org/10.3390/electronics10070826

Ilhan, H. O., & Amasyalı, M. F. (2014). Active Learning as a Way of Increasing Accuracy. *International Journal of Computer Theory and Engineering*, *6*(6), 460–465. https://doi.org/10.7763/ijcte.2014.v6.910

Jacob, N. (2005). Markov Processes and Applications. In *Finance*. https://doi.org/10.1002/9780470721872

Javaid, M., Haleem, A., Pratap Singh, R., & Suman, R. (2021). Significance of Quality 4.0 towards comprehensive enhancement in manufacturing sector. *Sensors International*, *2*, 100109. https://doi.org/10.1016/J.SINTL.2021.100109

Jiaqi, Y. (2010). *A Modeling Approach for Quality Inspection in Supply Chain Quality Management. July*.

Jo, D. U., Yun, S., & Choi, J. Y. (2022). How Much a Model be Trained by Passive Learning Before Active Learning? *IEEE Access*, *10*, 34677–34689. https://doi.org/10.1109/ACCESS.2022.3162253

Johnson, E. H., & Freeman, L. C. (1966). Elementary Applied Statistics: For Students in Behavioral Science. *Social Forces*, *44*(3). https://doi.org/10.2307/2575887

Johnson, T. L., Fletcher, S. R., Baker, W., & Charles, R. L. (2019). How and why we need to capture tacit knowledge in manufacturing: Case studies of visual inspection. *Applied Ergonomics*, *74*, 1–9. https://doi.org/10.1016/j.apergo.2018.07.016

Joshi, A. J., Porikli, F., & Papanikolopoulos, N. (2009). Multi-class active learning for image classification. *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, 2372–2379. https://doi.org/10.1109/CVPRW.2009.5206627

Kale, A. (2018). *A meta learning approach to one step active learning*. Medium. https://medium.com/@kaleajit27/apaperaday-week1-a-meta-learning-approach-to-one-step-active-learning-5ffea59099a2

Kiassat, C., Safaei, N., & Banjevic, D. (2014). Effects of operator learning on production output: A Markov chain approach. *Journal of the Operational Research Society*, *65*(12), 1814–1823. https://doi.org/10.1057/jors.2013.98

Kim, D., Seo, S. B., Yoo, N. H., & Shin, G. (2022a). A Study on Sample Size Sensitivity of Factory Manufacturing Dataset for CNN-Based Defective Product Classification. *Computation*, *10*(8). https://doi.org/10.3390/computation10080142

Kim, D., Seo, S. B., Yoo, N. H., & Shin, G. (2022b). A Study on Sample Size Sensitivity of Factory Manufacturing Dataset for CNN-Based Defective Product Classification. *Computation*, *10*(8). https://doi.org/10.3390/computation10080142

Kumar, P., & Gupta, A. (2020). Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey. *Journal of Computer Science and Technology*, *35*(4), 913–945. https://doi.org/10.1007/s11390-020-9487-4

Kunz, F. (2000). An Introduction to Temporal Difference Learning. *Seminar on Autonomous Learning Systems*, 21–22.

Leigh, J. M., Jackson, L., Dunnett, S., Lugo, H., Sharpe, R., Neal, A., & West, A. (2017). Modelling manufacturing processes using markov chains. *Safety and Reliability - Theory and Applications - Proceedings of the 27th European Safety and Reliability Conference, ESREL 2017*, 2497–2502. https://doi.org/10.1201/9781315210469-316

Lewis, F. L., & Liu, D. (2013). *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*.

Li, H., Liao, X., & Carin, L. (2009). Active learning for semi-supervised multi-task learning. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *1*, 1637–1640. https://doi.org/10.1109/ICASSP.2009.4959914

Li, J., & Meerkov, S. M. (2009). *Production systems engineering*. Springer.

Li, Y. (2017). *Deep Reinforcement Learning: An Overview*. https://doi.org/10.48550/arxiv.1701.07274

Lin, L. (2021). *Reinforcement Learning for Markov Decision Making in Env_Bellman_Q-learning_Q-Value Iteration*. https://blog.csdn.net/linli522362242/article/details/117889535

Liu, Z., Ding, H., Zhong, H., Li, W., Dai, J., & He, C. (2021). Influence Selection for Active Learning. *Proceedings of the IEEE International Conference on Computer Vision*, 9254–9263. https://doi.org/10.1109/ICCV48922.2021.00914

Luo, R., & Wang, X. (2020). Batch Active Learning with Two-Stage Sampling. *IEEE Access*, *8*, 46519–46528. https://doi.org/10.1109/ACCESS.2020.2979315

Luo, T., Kramer, K., Goldgof, D. B., Hall, L. O., Samson, S., Remsen, A., & Hopkins, T. (2005). Active Learning to Recognize Multiple Types of Plankton. In *Journal of Machine Learning Research* (Vol. 6).

Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., & Zhang, Y. (2018). Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sensing*, *10*(7). https://doi.org/10.3390/rs10071119

Mathwoks, A. (2019). Reinforcement Learning with MATLAB Understanding Training and Deployment. *Reinforcement Learning with MATLAB Understanding Training and Deployment*, 39. https://uk.mathworks.com/content/dam/mathworks/ebook/gated/reinforcement-learning-ebook-part3.pdf

MATLAB. (2020). *Reinforcement Learning Toolbox*.

Maulidi, I., Hayati, C., & Apriliani, V. (2022). *Optimal Raw Material Inventory Analysis Using Markov Decision Process with Policy Iteration Method*. *6*(3), 638–650.

Mausam, & Kolobov, A. (2012). Planning with markov decision processes: An AI perspective. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *17*, 1–203. https://doi.org/10.2200/S00426ED1V01Y201206AIM017

Mayer, C., & Timofte, R. (2020). Adversarial sampling for active learning. *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, 3060–3068. https://doi.org/10.1109/WACV45572.2020.9093556

Mayumu, N. (2022). *A survey of Deep learning at the Edge computing*. 10–13. https://doi.org/10.36227/techrxiv.19640265.v1

Michael, J. (2006). Where's the evidence that active learning works? *American Journal of Physiology - Advances in Physiology Education*, *30*(4), 159–167. https://doi.org/10.1152/advan.00053.2006

Mittal, S., Tatarchenko, M., Çiçek, Ö., & Brox, T. (2019a). *Parting with Illusions about Deep Active Learning*. *3*. http://arxiv.org/abs/1912.05361

Mittal, S., Tatarchenko, M., Çiçek, Ö., & Brox, T. (2019b). *Parting with Illusions about Deep Active Learning*. http://arxiv.org/abs/1912.05361

Moustapha, M., Marelli, S., & Sudret, B. (2022). Active learning for structural reliability: Survey, general framework and benchmark. *Structural Safety*, *96*, 102174. https://doi.org/10.1016/j.strusafe.2021.102174

Munjal, P., Hayat, N., Hayat, M., & Khan, J. S. and S. (2022). Towards Robust and Reproducible Active Learning Using Neural Networks. *CVPR*.

Munjal, P., Hayat, N., Hayat, M., Sourati, J., & Khan, S. (2020). *Towards Robust and Reproducible Active Learning using Neural Networks*. 223–232. https://doi.org/10.1109/cvpr52688.2022.00032

Nair, A., Pong, V., Dalal, M., Bahl, S., Lin, S., & Levine, S. (2018). *Visual Reinforcement Learning with Imagined Goals*. http://arxiv.org/abs/1807.04742

Nalubowa, M., Namango, S., Ochola, J., & Kizito Mubiru, P. (2021). Application of Markov chains in manufacturing systems: A review. *International Journal of Industrial Engineering and Operational Research (IJIEOR)*, *3*(1). http://www.ijieor.ir

Nashaat, M., Ghosh, A., Miller, J., Quader, S., Marston, C., & Puget, J. F. (2019). Hybridization of Active Learning and Data Programming for Labeling Large

Industrial Datasets. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 46–55. https://doi.org/10.1109/BigData.2018.8622459

Nguyen, H. T., Shin, N. R., Yu, G. H., Kwon, G. J., Kwak, W. Y., & Kim, J. Y. (2020). Deep learning-based defective product classification system for smart factory. *ACM International Conference Proceeding Series*, 80–85. https://doi.org/10.1145/3426020.3426039

Nguyen, V. L., Shaker, M. H., & Hüllermeier, E. (2022). How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, *111*(1), 89–122. https://doi.org/10.1007/s10994-021-06003-9

Ning, K. P., Zhao, X., Li, Y., & Huang, S. J. (2022). Active Learning for Open-set Annotation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2022-June*, 41–49. https://doi.org/10.1109/CVPR52688.2022.00014

Oliver, J. (2013). Hands-On Q-Learning with Python. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).

Özbay, E. (2022). An active deep learning method for diabetic retinopathy detection in segmented fundus images using artificial bee colony algorithm. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-022-10231-3

Paraschos, P. D., Koulinas, G. K., & Koulouriotis, D. E. (2020). Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. *Journal of Manufacturing Systems*, *56*, 470–483. https://doi.org/10.1016/J.JMSY.2020.07.004

Pontrandolfo, P., Gosavi, A., Okogbaa, O. G., & Das, T. K. (2002). Global supply chain management: A reinforcement learning approach. *International Journal of Production Research*, *40*(6), 1299–1317. https://doi.org/10.1080/00207540110118640

Pullar-Strecker, Z., Dost, K., Frank, E., & Wicker, J. (2022). Hitting the target: stopping active learning at the cost-based optimum. *Machine Learning*. https://doi.org/10.1007/s10994-022-06253-1

Ram Sagar. (2020, October 8). *What Is Deep Active Learning: Challenges and Applications*. https://analyticsindiamag.com/deep-active-learning-challenges-applications/

Ranganathan, H., Venkateswara, H., Chakraborty, S., & Panchanathan, S. (2018). Deep Active Learning for Image Classification. *2017 IEEE International Conference on Image Processing (ICIP)*. https://doi.org/10.1109/ICIP.2017.8297020

Ravichandiran, S. (2018a). *Elements of RL - Hands-On Reinforcement Learning with Python*. 318. https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788836524/1/ch01lvl1sec13/elements-of-rl

Ravichandiran, Sudharsan. (2018b). *Hands-On Reinforcement Learning with Python : Master Reinforcement and Deep Reinforcement Learning Using OpenAI Gym and TensorFlow*. Packt Publishing Ltd.

Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Gupta, B. B., Chen, X., & Wang, X. (2022). A Survey of Deep Active Learning. *ACM Computing Surveys*, *54*(9). https://doi.org/10.1145/3472291

Rieback, M. R., Crispo, B., & Tanenbaum, A. S. (2006). The evolution of RFID security. *IEEE Pervasive Computing*, *5*(1), 62–69. https://doi.org/10.1109/MPRV.2006.17

Roy, B. van, & Tsitsiklis, J. N. (2002). On Average Versus Discounted Reward Temporal-Difference Learning ∗. *Machine Learning*, *49*, 179–191.

Saleh, S. (2021). Deep Active Learning for Image Classification using Different Sampling Strategies. In *DEGREE PROJECT IN TECHNOLOGY*.

Sam, W., Zhang, Y., Long, X., Zhao, T., Wan, Y., Gao, L., Li, X., & Gao, Y. (2022). Semi-Supervised Defect Detection Method with Data-Expanding Strategy for PCB Quality Inspection. *Sensors 2022, Vol. 22, Page 7971*, *22*(20), 7971. https://doi.org/10.3390/S22207971

Sauer, A., Gramacy, R. B., & Higdon, D. (2022). Active Learning for Deep Gaussian Process Surrogates. *Technometrics*, 1–33. https://doi.org/10.1080/00401706.2021.2008505

Schmitt, J., Bönig, J., Borggräfe, T., Beitinger, G., & Deuse, J. (2020). Predictive model-based quality inspection using Machine Learning and Edge Cloud Computing. *Advanced Engineering Informatics*, *45*(May), 101101. https://doi.org/10.1016/j.aei.2020.101101

See, J. E. (2012). *SANDIA REPORT Visual Inspection: A Review of the Literature*. http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

See, J. E. (2015). Visual Inspection Reliability for Precision Manufactured Parts. *Human Factors*, *57*(8), 1427–1442. https://doi.org/10.1177/0018720815602389

Sener, O., & Savarese, S. (2018). Active learning for convolutional neural networks: A core-set approach. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Settles, B. (2009). *Active Learning Literature Survey*. *January*.

Shafiq, M., & Gu, Z. (2022a). Deep Residual Learning for Image Recognition: A Survey. *Applied Sciences (Switzerland)*, *12*(18), 1–43. https://doi.org/10.3390/app12188972

Shafiq, M., & Gu, Z. (2022b). Deep Residual Learning for Image Recognition: A Survey. In *Applied Sciences (Switzerland)* (Vol. 12, Issue 18). MDPI. https://doi.org/10.3390/app12188972

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Sharpe, R., Banwell, G., … P. C.-2014 I. 16th, & 2014, undefined. (2012). Sensor-enabled PCBs to aid right first time manufacture through defect prediction. *Ieeexplore.Ieee.Org*. https://ieeexplore.ieee.org/abstract/document/7028396/

Shirai, T. (2014). *Finite Markov Chains and Markov Decision Processes*. 189–206. https://doi.org/10.1007/978-4-431-55060-0_15

Sigaud, Olivier., & Buffet, Olivier. (2010). *Markov decision processes in artificial intelligence : MDPs, beyond MDPs and applications*. ISTE.

Solic, P., Rozic, N., & Marinovic, S. (2009, June 8). RFID-based visitors modeling for galleries using Markov model. *International Conference on Telecommunications*. https://www.researchgate.net/publication/224579097_RFID-based_visitors_modeling_for_galleries_using_Markov_model

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-October*, 843–852. https://doi.org/10.1109/ICCV.2017.97

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. https://books.google.com.my/books?hl=en&lr=&id=uWV0DwAAQBAJ&oi=fnd&pg=PR7&dq=Sutton,+R.+S.,+and+Barto,+A.+G.+(2018).+Reinforcement+learning:+An+introduction.+MIT+press.&ots=miuIp4W4o0&sig=kj-FbW9zaEBneF-QDM-SOghIPuQ

Szepesvári, C. (2010a). Algorithms for reinforcement learning. In *Synthesis Lectures on Artificial Intelligence and Machine Learning* (Vol. 9). https://doi.org/10.2200/S00268ED1V01Y201005AIM009

Szepesvári, C. (2010b). Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *9,* 1–89. https://doi.org/10.2200/S00268ED1V01Y201005AIM009

Takezoe, R., Liu, X., Mao, S., Chen, M. T., Feng, Z., Zhang, S., & Wang, X. (2022). *Deep Active Learning for Computer Vision: Past and Future*. 1–18. http://arxiv.org/abs/2211.14819

Tang, M., Luo, X., & Roukos, S. (2002). Active Learning for Statistical Natural Language Parsing. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (Pp. 120-127).*

Tang, Y.-P., Li, G.-X., & Huang, S.-J. (2019). *ALiPy: Active Learning in Python*. http://arxiv.org/abs/1901.03802

Tanwar, S. (2019). Bellman Equation and dynamic programming. *Analytics Vidhya*. https://medium.com/analytics-vidhya/bellman-equation-and-dynamic-programming-773ce67fc6a7

Tanwar, S. (2020). *What is reinforcement learning ? How reinforcement learning differ from other machine learning algorithms ? Elements of RL*. 1–5.

Tersine, R. (1994). *Principles of inventory and materials management*. http://repo.unikadelasalle.ac.id/index.php?p=show_detail&id=4449&keywords=

Tesauro, G. (1992). Practical issues in temporal difference learning. *Machine Learning*, *8*(3–4), 257–277. https://doi.org/10.1007/bf00992697

Tochukwu, C., & Hyacinth, I. (2015). Agent Based Markov Chain for Job Shop Scheduling and Control: Review of the Modeling Technigue. In *IJISET-International Journal of Innovative Science, Engineering & Technology* (Vol. 2, Issue 3). www.ijiset.com

Tsvigun, A., Shelmanov, A., Kuzmin, G., Sanochkin, L., Larionov, D., Gusev, G., Avetisian, M., & Zhukov, L. (2022). Towards Computationally Feasible Deep Active Learning. *Findings of the Association for Computational Linguistics: NAACL 2022 - Findings*, 1198–1218. https://doi.org/10.18653/v1/2022.findings-naacl.90

van Wesel, P., & Goodloe, A. E. (2017). Challenges in the Verification of Reinforcement Learning Algorithms. *NASA Langely Research Center*, *1*(1), 1–24. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170007190.pdf

Vatsal. (2022). *Active Learning in Machine Learning Explained (Intuition and Implementation of an Active Learning Pipeline in Python)*. Medium. https://towardsdatascience.com/active-learning-in-machine-learning-explained-777c42bd52fa

Wallin, C., Rungtusanatham, M. J., & Rabinovich, E. (2006). What is the 'right' inventory management approach for a purchased item? *International Journal of Operations and Production Management*, *26*(1), 50–68. https://doi.org/10.1108/01443570610637012

Wang, K., Zhang, D., Li, Y., Zhang, R., & Lin, L. (2017). Cost-Effective Active Learning for Deep Image Classification. *IEEE Transactions on Circuits and Systems for Video Technology*, *27*(12). https://doi.org/10.1109/TCSVT.2016.2589879

Wang, T., Chen, S., & Jia, R. (2021). *One-Round Active Learning*. http://arxiv.org/abs/2104.11843

Watkins, C. J. C. H., & Dayan, P. (1992). Technical Note: Q-Learning. *Machine Learning*, *8*(3), 279–292. https://doi.org/10.1023/A:1022676722315

Weinstein, R. (2005). RFID: A technical overview and its application to the enterprise. *IT Professional*, *7*(3), 27–33. https://doi.org/10.1109/MITP.2005.69

Winston, W. L. , & Goldberg, J. B. (2004). *Operations research: applications and algorithms* (Vol. 3). Cengage Learning. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Winston%2C+W.+L.%2C+%26+Goldberg%2C+J.+B.+%282004%29.+Operations+research%3A+applications+and+algorithms+%28Vol.+3%29.+Belmont%3A+Thomson+Brooks%2FCole.&btnG=

Xie, Y., Tomizuka, M., & Zhan, W. (2021). *Towards General and Efficient Active Learning*. http://arxiv.org/abs/2112.07963

Xing, E. (2008). *Reinforcement learning 2 Markov Decision Process (MDP)*. 1–22.

Yang, A. X. (2020). Markov chain and its applications. *Applied Data Analytics: Principles and Applications*, *March*, 1–15. https://doi.org/10.1201/9781003337225-1

Ye, Y., Grossmann, I. E., Pinto, J. M., & Ramaswamy, S. (2019). Modeling for reliability optimization of system design and maintenance based on Markov chain theory. *Computers and Chemical Engineering*, *124*, 381–404. https://doi.org/10.1016/j.compchemeng.2019.02.016

Yehuda, O., Dekel, A., Hacohen, G., & Weinshall, D. (2022). *Active Learning Through a Covering Lens*. *NeurIPS*. http://arxiv.org/abs/2205.11320

Zaghdoud, R., Boukthir, K., Hamdani, T. M., & Alimi, A. M. (2022). Deep Active Learning Approach for Traffic Sign and Panel Guide Arabic-Latin Text Content Annotation in Natural Scene Images. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4127046

Zhan, X., Dai, Z., Wang, Q., Li, Q., Xiong, H., Dou, D., & Chan, A. B. (2022). *Pareto Optimization for Active Learning under Out-of-Distribution Data Scenarios*. http://arxiv.org/abs/2207.01190

Zhan, X., Wang, Q., Huang, K., Xiong, H., Dou, D., & Chan, A. B. (2022a). *A Comparative Survey of Deep Active Learning*. http://arxiv.org/abs/2203.13450

Zhan, X., Wang, Q., Huang, K., Xiong, H., Dou, D., & Chan, A. B. (2022b). *A Comparative Survey of Deep Active Learning*. http://arxiv.org/abs/2203.13450

Zhan, X., Wang, Q., Huang, K., Xiong, H., Dou, D., & Chan, A. B. (2022c). *A Comparative Survey of Deep Active Learning*. http://arxiv.org/abs/2203.13450

Zhang, H., Chen, J. C., & Zhang, K. (2014). RFID-based localization system for mobile robot with Markov Chain Monte Carlo. *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education - 'Engineering Education: Industry Involvement and Interdisciplinary Trends', ASEE Zone 1 2014*. https://doi.org/10.1109/ASEEZONE1.2014.6820672

Zhang, X. Y., Wang, S., & Yun, X. (2015). Bidirectional active learning: A two-way exploration into unlabeled and labeled data set. *IEEE Transactions on Neural*

Networks and Learning Systems, *26*(12), 3034–3044. https://doi.org/10.1109/TNNLS.2015.2401595

Zhou, L., Zhang, L., & Konz, N. (2022). Computer Vision Techniques in Manufacturing. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *2021*, 0–13. https://doi.org/10.1109/TSMC.2022.3166397

APPENDIX A

Table 2.1 Summary of research gap from the literature

Authors	Title	Research Gaps
(Belusso et al., 2016)	A Study of Petri Nets, Markov Chains and Queueing Theory as Mathematical Modelling Languages Aiming at the Simulation of Enterprise Application Integration Solutions: A First Step	
(Hu et al., 2019)	Effect of the manufacturer quality inspection policy on the supply chain decision-making and profits	
(Nalubowa et al., 2021)	Application of Markov chains in manufacturing systems: A review	
(Buffett, 2005)	A Markov model for inventory level optimization in supply-chain management	
(Shirai, 2014)	Finite Markov Chains and Markov Decision Processes	
(H. Zhang et al., 2014)	RFID-based localization system for mobile robot with Markov Chain Monte Carlo	
(Solic et al., 2009)	RFID-based visitors modelling for galleries using Markov model	
(Yang, 2020)	Markov chain and its applications	
(Mausam & Kolobov, 2012)	Planning with Markov Decision Processes	

(Sigaud & Buffet, 2010)	Markov Decision Processes in Artificial Intelligence	
(Brachman et al., 2009)	Markov Logic: An Interface Layer for Artificial Intelligence	

Table 2.3 Review of research gap in the field of Quality inspection

Authors	Title	Research Gaps
(Schmitt et al., 2020)	Predictive model-based quality inspection using Machine Learning and Edge Cloud Computing	
(Sam et al., 2022)	Semi-Supervised Defect Detection Method with Data-Expanding Strategy for PCB Quality Inspection	
(K.-H. Huang, 2021a)	Deep AL – Deep Active Learning Library using Python	
(Ram Sagar, 2020)	Applications and Challenges of Deep Active Learning	
(Ren et al., 2022)	Survey on Deep Active Learning	
(Zhan, Wang, et al., 2022a)	A Comparative Survey of Deep Active Learning	
(Takezoe et al., 2022)	Deep Active Learning for Computer Vision: Past and Future	
(Mittal et al., 2019a)	Parting with Illusions about Deep Active Learning	
(Geifman & El-Yaniv, 2017)	Deep Active Learning over the Long Tail	
(Fang et al., 2017a)	Learning how to active learn: A deep reinforcement learning approach	
(Geifman & El-Yaniv, n.d.)	Deep Active Learning with a Neural Architecture Search	

(Hemmer et al., n.d.)	DEAL: Deep Evidential Active Learning for Image Classification	
(Gal et al., 2017)	Deep Bayesian active learning with image data	
(Beck et al., 2021a)	Effective Evaluation of Deep Active Learning on Image Classification Tasks	
(Saleh, 2021)	Deep Active Learning for Image Classification using Different Sampling Strategies	
(Adrian et al., 2021)	Best Practices in Pool Based Active Learning for Image Classification	
(Gudovskiy et al., 2020)	Deep active learning for biased datasets via fisher kernel self-supervision	
(Agarwal, 2022)	Active and Incremental Deep learning	
(Özbay, 2022)	An active deep learning method for diabetic retinopathy detection in segmented fundus images using artificial bee colony algorithm	
(Zaghdoud et al., 2022)	Deep Active Learning Approach for Traffic Sign and Panel Guide Arabic-Latin Text Content Annotation in Natural Scene Images	
(Tsvigun et al., 2022)	Towards Computationally Feasible Deep Active Learning	
(Ranganathan et al., 2018)	Deep Active Learning for Image Classification	
(Sauer et al., 2022)	Active Learning for Deep Gaussian Process Surrogates	
(Ning et al., 2022)	Active Learning for Open-set Annotation	
(Michael, 2006)	Where's the evidence that active learning works?	
(Mittal et al., 2019b)	Parting with Illusions about Deep Active Learning	
(Hemmer et al., 2020)	DEAL: Deep Evidential Active Learning for Image Classification	

(Geifman & El-Yaniv, n.d.)	Deep Active Learning with a Neural Architecture Search	
(W. Huang et al., 2022)	Deep active learning with Weighting filter for object detection	
(Beck et al., 2021b)	Effective Evaluation of Deep Active Learning on Image Classification Tasks	
(Adrian et al., 2021)	Best Practices in Pool Based Active Learning for Image Classification	
(Zaghdoud et al., 2022)	Deep Active Learning Approach for Traffic Sign and Panel Guide Arabic-Latin Text Content Annotation in Natural Scene Images	

Active Learning

(Contardo et al., 2017)	A Meta-Learning Approach to One-Step Active-Learning	
	Does active learning work? A review of the research	
(Jo et al., 2022)	How Much a Model be Trained by Passive Learning Before Active Learning?	
(X. Y. Zhang et al., 2015)	Bidirectional active learning: A two-way exploration into unlabeled and labeled data set	
(Anahideh et al., 2022)	Fair active learning	
(Settles, 2009)	Active Learning Literature Survey	

(Liu et al., 2021)	Influence Selection for Active Learning	
(S. Chen et al., 2021)	Zero-Round Active Learning	
(T. Wang et al., 2021)	One-Round Active Learning	
(Mayer & Timofte, 2020)	Adversarial sampling for active learning	
(Yehuda et al., 2022)	Active Learning Through a Covering Lens	
(Xie et al., 2021)	Towards General and Efficient Active Learning	
(Aggarwal et al., 2022)	Optimizing Active Learning for Low Annotation Budgets	
(Aurobindo Munagala et al., n.d.)	CLACTIVE: EPISODIC MEMORIES FOR RAPID ACTIVE LEARNING	
(H. Li et al., 2009)	Active learning for semi-supervised multi-task learning	
(R. Luo & Wang, 2020)	Batch Active Learning with Two-Stage Sampling	
(Ilhan & Amasyalı, 2014)	Active Learning as a Way of Increasing Accuracy	
(Munjal et al., 2020)	Towards Robust and Reproducible Active Learning using Neural Networks	
(V. L. Nguyen et al., 2022)	How to measure uncertainty in uncertainty sampling for active learning	
(Desai et al., 2020)	An adaptive supervision framework for active learning in object detection	

(Moustapha et al., 2022)	Active learning for structural reliability: Survey, general framework and benchmar	
(Desai et al., 2019)	An Adaptive Supervision Framework for Active Learning in Object Detection	
(Kumar & Gupta, 2020)	Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey	
(Pullar-Strecker et al., 2022)	Hitting the target: stopping active learning at the cost-based optimum	
(Nashaat et al., 2019)	Hybridization of Active Learning and Data Programming for Labeling Large Industrial Datasets	
(Zhan, Dai, et al., 2022)	Pareto Optimization for Active Learning under Out-of-Distribution Data Scenarios	
(Atighehchian et al., 2020)	Bayesian active learning for production, a systematic study and a reusable library	
(Haussmann et al., 2019)	Scalable Active Learning for Autonomous Driving	
(Özbay, 2022)	An active deep learning method for diabetic retinopathy detection in segmented fundus images using artificial bee colony algorithm	

(Fang et al., 2017b)	Learning how to Active Learn: A Deep Reinforcement Learning Approach	
(Joshi et al., 2009)	**Multi-class active learning for image classification**	
(Bengar et al., 2021)	When Deep Learners Change Their Mind: Learning Dynamics for Active Learning	
(Caramalau et al., 2023)	MoBYv2AL: Self-supervised Active Learning for Image Classification	

Table 2.3 Review of active learning libraries

Authors	**Library**	**Publication**	**Link**
Kuan-Hao Huang (K.-H. Huang, 2021b)	DeepAL: Deep Active Learning in Python		https://github.com/ej0cl6/deep-active-learning

	Deep Active Learning with Pytorch		https://github.com/cure-lab/deep-active-learning

Riashat Islam (Gal et al., 2017)	Deep-Bayesian-Active-Learning	Deep Bayesian Active Learning with Image Data	https://github.com/Riashat/Deep-Bayesian-Active-Learning
Hung-Tu Chen	Pytorch Deep Q Learning		https://github.com/hungtuchen/pytorch-dqn
(Y.-P. Tang et al., 2019)	ALiPy: Active Learning in Python		

Deep Active learning for Image Segmentation

Mélanie Lubrano	Deep Active Learning for Myelin Segmentation on Histology Data		https://github.com/neuropoly/deep-active-learning
	Active Deep Learning for Medical Imaging Segmentation		https://github.com/marc-gorriz/CEAL-Medical-Image-Segmentation

Deep Active Learning for Natural Language Processing

Aditya Siddhant	Active-NLP		https://github.com/asiddhant/Active-NLP

Table 2.2 Review of research gap in Reinforcement Learning

Authors	**Title**	**Research Gaps**
(Maulidi et al., 2022)	Optimal Raw Material Inventory Analysis Using Markov Decision Process with Policy Iteration Method	
(Habib, 2019)	Hands-on Q-learning with Python: practical Q-learning with OpenAI Gym, Keras, and TensorFlow	
(Roy & Tsitsiklis, 2002)	On Average Versus Discounted Reward Temporal-Difference Learning	
(Tesauro, 1992)	Practical issues in temporal difference learning	
(Kunz, 2000)	An Introduction to Temporal Difference Learning	
(Bertsekas, 2010)	Pathologies of temporal difference methods in approximate dynamic programming	
(Tanwar, 2019)	Bellman Equation and dynamic programming	
(Bertsekas, 2012)	Dynamic Programming and Optimal Control (2 Vol Set)	
(Azcue & Muler, 2014)	Stochastic Optimization in Insurance: A Dynamic Programming Approach	
(Lewis & Liu, 2013)	Reinforcement Learning and Approximate Dynamic Programming for Feedback Control	
(Arulkumaran et al., 2017)	Deep Reinforcement Learning: A brief survey	
(Fenjiro & Benbrahim, 2018)	Deep Reinforcement Learning Overview of the state of the Art.	
(Watkins & Dayan, 1992)	Technical Note: Q-Learning	
(Habib, 2019)	Hands-on Q-learning with Python: practical Q-learning with OpenAI Gym, Keras, and TensorFlow	
(Doltsinis et al., 2014)	**An MDP model-based reinforcement learning approach for production station ramp-up**	

	optimization: Q-learning analysis	
(Oliver, 2013)	Hands-On Q-Learning with Python	
(Sutton & Barto, 2018)	Reinforcement learning: An introduction	
(Adams & Elements, n.d.)	Reinforcement Learning	
(MATLAB, 2020)	Reinforcement Learning Toolbox	
(Szepesvári, 2010b)	Algorithms for reinforcement learning	
(Ravichandiran, 2018b)	Hands-On Reinforcement Learning with Python: Master Reinforcement and Deep Reinforcement Learning Using OpenAI Gym and TensorFlow.	
(Gosavi, 2019)	A Tutorial for Reinforcement Learning	
(Y. Li, 2017)	Deep Reinforcement Learning: An Overview	
(Nair et al., 2018)	Some Recent Applications of Reinforcement Learning	
(Nair et al., 2018)	Visual Reinforcement Learning with Imagined Goals	
(Arulkumaran et al., 2017)	A Brief Survey of Deep Reinforcement Learning	
(Mathwoks, 2019)	Reinforcement Learning with MATLAB Understanding Training and Deployment	
(Tanwar, 2020)	**What is reinforcement learning? How reinforcement learning differs from other machine learning algorithms? Elements of RL**	
(van Wesel & Goodloe, 2017)	Challenges in the Verification of Reinforcement Learning Algorithms	
(Hulbert et al., 2021)	EasyRL: A Simple and Extensible Reinforcement Learning Framework	
(Gronauer & Diepold, 2022)	Multi-agent deep reinforcement learning: a survey	
(Pontrandolfo et al., 2002)	Global supply chain management: A reinforcement learning approach	

(Lewis & Liu, 2013)	Reinforcement Learning and Approximate Dynamic Programming for Feedback Control	
(Xing, 2008)	Reinforcement learning 2 Markov Decision Process (MDP)	
(Fenjiro & Benbrahim, 2018)	Deep Reinforcement Learning Overview of the state of the Art.	
(Abhishek & Biswas, 2018)	Reinforcement learning with Open AI, TensorFlow and Keras Using Python	
(Ravichandiran, 2018a)	Elements of RL - Hands-On Reinforcement Learning with Python	
(M. Chen et al., 2021)	Deep Reinforcement Learning for the Computation Offloading in MIMO-based Edge Computing environment	
(Barde et al., 2019)	Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks	
(Paraschos et al., 2020)	Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures	
(Bone & Dragićević, 2010)	Simulation and validation of a reinforcement learning agent-based model for multi-stakeholder forest management	
(Doltsinis et al., 2014)	An MDP model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis	
(Elmachtoub et al., 2020)	Decision trees for decision-making under the predict-then-optimize framework	
(Agerskans, 2019)	A Framework for Achieving Data-Driven Decision Making in Production Development	

APPENDIX B

APPENDIX C

LIST OF PUBLICATIONS

Conferences

Mani, A., Yaacob, S., Krishnan, P., AT Othman, & MA Abas (2016). RFID based markov chain model for automotive supply chain, Conference on Language, Education Engineering and Technology (COLEET) 2016.

Journals

Mani, A., Bakar, S. A., Krishnan, P., & Yaacob, S. (2021, November). Markov Decision Process approach in the estimation of raw material quality in incoming inspection process. In Journal of Physics: Conference Series (Vol. 2107, No. 1, p. 012025). IOP Publishing. **DOI** 10.1088/1742-6596/2107/1/012025

Mani, A., Bakar, S. A., Krishnan, P., & Yaacob, S. (2021, November). Comparison of optimized Markov Decision Process using Dynamic Programming and Temporal Differencing–A reinforcement learning approach. In Journal of Physics: Conference Series (Vol. 2107, No. 1, p. 012026). IOP Publishing. **DOI** 10.1088/1742-6596/2107/1/012026

Mani, A., Bakar, S. A., Krishnan, P., & Yaacob, S. (2021, November). Categorization of material quality using a model-free reinforcement learning algorithm. In Journal of Physics: Conference Series (Vol. 2107, No. 1, p. 012027). IOP Publishing. **DOI** 10.1088/1742-6596/2107/1/012027