TALLINN UNIVERSITY OF TECHNOLOGY
SCHOOL OF ENGINEERING
Department of Mechanical and Industrial Engineering

MACHINE LEARNING-BASED ANALYSIS OF PRODUCTION MATRICES FOR ENHANCING PRODUCTIVITY IN SMALL AND MEDIUM-SIZED ENTERPRISES (SMES)

MASINÕPPEL PÕHINEV TOOTMISMÕÕDIKUTE ANALÜÜS VÄIKESE JA KESKMISE SUURUSEGA ETTEVÕTETE (VKE) TOOTLIKKUSE SUURENDAMISEKS

MASTER THESIS

Student: Nilam Basel

Student code: 214600MARM

Supervisor: Jüri Majak, Full tenured professor

Tallinn 2023

(*On the reverse side of title page*)

AUTHOR’S DECLARATION

Hereby I declare, that I have written this thesis independently.

No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

“....20...” .05.................. 2023

Author: Nilam Basel

/signature /

Thesis is in accordance with terms and requirements

“...22....” ....05........... 2023….

Supervisor: Jüri Majak

/signature/

Accepted for defence

“.......”...................20… .

Chairman of theses defence commission: ....

/name and signature/

Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Nilam Basel,

1. grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis Machine Learning-based Analysis of Production Metrics for Enhancing Productivity in Small and Medium-sized Enterprises (SMEs),

supervised by Jüri Majak,

1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2. I am aware that the author also retains the rights specified in clause 1 of the non- exclusive licence.

3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

Department of Mechanical and Industrial Engineering

20.05.2023

[1] *The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.*

THESIS TASK

Student: Nilam Basel, 214600MARM

Study programme: MARM06/18 - Industrial Engineering and Management

main speciality: Industrial Engineering

Supervisor(s): Jüri Majak, Full tenured professor

Thesis topic:

(in English) *Machine Learning-based Analysis of Production Metrics for Enhancing Productivity in Small and Medium-sized Enterprises (SMEs)*

(in Estonian) *Masinõppel põhinev tootmismõõdikute analüüs väikese ja keskmise suurusega ettevõtete (VKE) tootlikkuse suurendamiseks.*

Thesis main objectives:

1. To overview and evaluation of the Smart Manufacturing enabled by artificial Intelligence in SMEs.
2. To understand the application of selected ML methods and tools for data modelling.
3. To compare and analyse the results obtained by different ML methods.

Thesis tasks and time schedule:

No	Task description	Deadline
1.	Selection of topic, research, and literature review draft	15.03
2.	Methodology, model training, and model evaluation	25.04
3.	Analysis of the obtained results, formatting, and corrections	01.05

Language: English **Deadline for submission of thesis:** 22.05.2023

Student: Nilam Basel 22.05.2023

Supervisor: Jüri Majak 22.05.2023

Head of study programme: Prof. Kristo Karjust 22.05.2023

Terms of thesis closed defence and/or restricted access conditions to be formulated on the reverse side

PREFACE

This master thesis is the result of my research on machine learning-based analysis of production matrices for enhancing productivity in small and medium-sized enterprises (SMEs). The idea for this thesis was born out of my interest in the field of industrial engineering and management, and my desire to explore how machine learning techniques can be used to optimize production processes in SMEs.

The main objective of this thesis is to provide an overview and evaluation of smart manufacturing enabled by artificial intelligence in SMEs. To achieve this objective, I have conducted a thorough literature review on the topic, analysed production matrices data collected from a machine in SMEs using traditional machine learning algorithms, and developed data-driven insights and prediction models for optimizing production productivity.

Throughout this thesis, I have highlighted the need for data-driven insights and prediction models in the manufacturing industry, particularly for SMEs with limited resources. By leveraging data analysis, big data analytics, and machine learning techniques, SMEs can improve their productivity and competitiveness in the market.

I would like to express my gratitude to my supervisor Jüri Majak for his guidance and support throughout this research. His expertise in the field of industrial engineering and management has been invaluable to me. I would also like to thank all the participants who provided me with their valuable insights during this research.

Finally, I hope that this thesis will serve as a useful resource for researchers, practitioners, and students interested in exploring how machine learning techniques can be used to optimize production processes in SMEs.

Keywords: Smart Manufacturing, Small and Medium-sized Enterprises, Machine Learning, Artificial Neural Networks, Productivity.

TABLE OF CONTENTS

LIST OF THE FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

SMEs	Small and Medium-sized Enterprises
ML	Machine Learning
AI	Artificial Intelligence
CPS	Cyber-Physical Systems
IOT	Internet of Things
IIoT	Industrial Internet of Things
SM	Smart Manufacturing
SMS	Smart Manufacturing System
OEE	Overall Equipment Efficiency
DT	Decision Tree
NN	Neural Network
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
RMSE	Root Mean Square Error
MSE	Mean Square Error
CSV	Comma Separated Value
3D	3-Dimensional

1. INTRODUCTION

The integration of digital technology into manufacturing processes has caused a paradigm change in the industry, altering production systems on multiple tiers. Smart Manufacturing concepts have played a critical role in improving the performance of manufacturing systems in terms of quality, cost efficiency, adaptability, and decision-making capacities. One significant strategy that is gaining popularity in this area is the use of statistical analysis, as opposed to machine learning (ML), to satisfy the changing needs of the dynamic industrial environment. Manufacturing SMEs can get various advantages by adopting statistical analysis solutions. Notably, statistical analysis excels at dealing with high-dimensional but knowledge-poor data, which is typical in manufacturing contexts. Furthermore, patterns shown from current data may be used to forecast the behaviour of industrial systems, providing decision assistance and system enhancement [1].

The main goal of this thesis is to undertake an in-depth investigation and analysis of the adoption of cost-effective statistical analysis solutions by small and medium-sized enterprises (SMEs) in the manufacturing sector. According to different research [2, 3, 4, 5, 6, 7, 8] made in the context of SMEs, SMEs face various challenges, including limited financial resources, a lack of analytical knowledge, difficulties in risk assessment and mitigation, and restricted access to comprehensive data. Additionally, traditional commercial analytics tools are often considered too expensive for SMEs. Therefore, there is a critical need for a systematic approach to design, develop, and integrate user-friendly, off-the-shelf, and cost-effective statistical analysis solutions. The objective of this thesis is to learn, understand and compare different AI tools for production data modelling in SMEs. To achieve this goal, a brief exploration of the evolving landscape of data analytics and machine learning (ML) in SMEs is conducted. Additionally, the potential applications of ML in enhancing production performance are examined, along with the identification of affordable, user-friendly ML methods suitable for SMEs.

Through a comprehensive literature review and empirical analysis, this thesis aims to bring insights into how SMEs can effectively leverage data analytics and ML for their growth and competitiveness in the era of Industry 4.0. The focus area of the study is explained as follows:

1. Overview and evaluation of the Smart Manufacturing enabled by artificial Intelligence in SMEs.

A comprehensive literature review will be conducted to examine the advancements, trends, and barriers associated with data analytics and ML adoption in SMEs.

2. Application of selected ML methods and tools for data modelling

Investigating the application of ML algorithms for analysing production data, identifying valuable insights, and predicting future trends will provide valuable guidance for SMEs seeking to enhance their productivity.

3. Comparison and analysis of the results obtained by different ML methods

Evaluating existing ML techniques, exploring low-cost alternatives, and providing recommendations for affordable and accessible ML tools and platforms will empower SMEs to leverage ML effectively without substantial financial burdens.

To achieve the goals, this study is divided into sections and subsections. The second section of this paper dives deep into the literature reviews and bring out research and their finding while relating to the real use case. This section provides insights about smart manufacturing in SEMs, different production matrices in manufacturing, ML, their types and how ML is making an impact on smart manufacturing. The third section layout the foundation for the case study of an application of ML using data from SMEs with the use of MATLAB as a platform for model training, followed by the application of different machine learning methods. The results of the model training are shown in the fourth section which helps to uncover the final research question and compile a conclusion. A comprehensive summary of the research findings and further work needed to improvise and implement the findings are provided at the end of the paper.

2. LITERATURE REVIEW

This section of the thesis provides brief background knowledge and the work done in the field of smart manufacturing systems, industrial data, analytics, and uses of the data to enhance decision-making in various industries.

2.1 Smart Manufacturing System (SMS)

Industrialization, which had advanced in the early 1900s, took on new significance in recent years with Industry 4.0. It is intended in this context to provide internet and object communication, accelerate mass production, gain data and information exchange, reduce raw material, resource, and energy consumption, incorporate robots into the manufacturing process, improve occupational health and safety, and integrate the virtual world into the manufacturing process [9].

As industries are rapidly transitioning to become smarter with the integration of sensors, smart devices, intra, and inter-device communication while capturing as much data as possible throughout the product lifecycle, data has become increasingly prominent for all business stages. With the rise in usages of Cyber-Physical Systems (CPS), it is increasingly getting easier and easier to implement SM concepts and technologies associated with it [9, 10]. Cyber-Physical Systems (CPS) are groups of cooperating computational entities that have a close relationship with the physical environment and its ongoing activities. They also provide and utilise both data processing and data access services that are offered online. CPS is a key technology for achieving Smart Manufacturing in the manufacturing sector, and it is being researched in conjunction with cloud, IoT, and big data technologies [10].

Smart manufacturing is often described using different terms such as "Digital manufacturing" and "Intelligent Manufacturing". Despite the different terms being used, the core concept of all three terms is the same concept of communication and computing technologies that enable all players in the value chain of products to be digitally connected and data analytics-driven at the supply chain, enterprise, and shop floor levels, achieving intelligent coordination for demand and supply matching, faster time to market, mass customization, and cost benefits [11]. This success of implementing smart manufacturing relies on the availability of smart IOT devices, wireless networking infrastructure and the resources for integrating such technologies.

Kusiak, Andrew (2017) has broken down the components of smart manufacturing to make it better and understand the basic components of SM [11].

- Smart Sensors and Processors
- Industrial Internet of Things (IIOT)
- Data Analytics and Cloud Computing
- Material Informatics
- Zero-waste Manufacturing
- Additive Manufacturing/3D Printing
- Cyber Security for Manufacturing
- International standards

These are the essential components to be listed but these components of the SM also possess various subcomponents of their own. A more details breakdown of the SM components can be visualized as [12]:

Figure 2.1.1 A schematic of smart manufacturing components [12]

As the progress of industries across many domains started to capitalize on the revolution of Industry especially Industry 4.0, factories are becoming smarter, and the production process has been faster and faster [12]. From robotization to smart machines that are cable of performing various tasks while providing constant feedback to machines that are cable of recovering by themself, smart factories have haven growing rapidly. Starting with the data documentation in papers and manufacturing realized by handcraft, with the introduction of electronic computers rapid development of information technology, the development of network communication like local area network (LAN), TCP/IP, World Wide Web (WWW) the advancement was already in rapid use and showing a hug potential change in the manufacturing. The result of the Appling such technology in m manufacturing saw the rise of computer-integrated manufacturing (CIM), computer-aided design (CAD), computer-aided manufacturing (CAM), machine execution system (MES), and enterprise resource planning (ERP) [13]. In the past decade, the growth of newer information technologies such as IIOT, big data analytics, cloud computing, and artificial intelligence is leading to new series of manufacturing concepts such as cyber-physical systems, manufacturing grids, cloud manufacturing, and intelligent manufacturing [13].

This new series of manufacturing concepts kept growing and newer concepts are being arising. Manufacturing industries of the future are moving toward sustainable manufacturing and resilient manufacturing with the usage of technologies. Because of its impact on the environment and future generations, sustainable manufacturing is a hot topic today. By reducing waste, energy consumption, and carbon emissions, digital manufacturing can help achieve sustainability goals. Circular economy practices are an important aspect of sustainable manufacturing concerning digital manufacturing. Manufacturers can design and produce products with longer lifetimes, reduce material waste, and facilitate material reuse or recycling by incorporating digital technologies such as 3D printing and advanced analytics. Integration of digital technologies with circular economy principles can also help to optimize resource consumption and reduce manufacturing's negative environmental impacts [14].

Energy efficiency is another important topic in sustainable manufacturing that is related to digital manufacturing. Intelligent automation and optimization algorithms, for example, can help to reduce energy consumption in manufacturing processes. Using predictive maintenance algorithms and machine learning techniques, for example, can help to optimize machine performance, lowering energy consumption and operational costs. Furthermore, incorporating renewable energy sources such as solar and wind power can aid in lowering the carbon footprint of manufacturing operations [15].

Finally, sustainable supply chain management is an important topic in digital manufacturing and sustainable manufacturing. Blockchain and Internet of Things (IoT) technologies, for example, can help to improve supply chain transparency and traceability. Manufacturers can identify areas for improvement, such as reducing transportation emissions and optimizing inventory levels, by tracking the origin and movement of raw materials. Furthermore, digital technologies can aid in the improvement of supplier relationships, leading to more sustainable practices throughout the supply chain [16].

Smart manufacturing systems, which incorporate advanced technologies such as the Internet of Things (IoT), artificial intelligence (AI), and cloud computing into manufacturing operations, rely heavily on resilient manufacturing. The ability of manufacturing systems to withstand disruptions and recover quickly from unexpected events such as natural disasters, equipment failures, or supply chain disruptions is referred to as resilience. Resilient manufacturing, in the context of smart manufacturing, entails the use of digital technologies to improve the flexibility, agility, and adaptability of manufacturing systems [17].

Predictive maintenance is one method for achieving resilient manufacturing in smart manufacturing systems. The use of AI and IoT technologies to monitor the condition of equipment and predict when maintenance is required is known as predictive maintenance. This method can aid in the prevention of unexpected equipment failures and the reduction of downtime, thereby increasing the resilience of manufacturing systems. Furthermore, digital twin technology can be used to simulate the behaviours of manufacturing systems and identify potential vulnerabilities, allowing for proactive risk mitigation and resilience measures to be implemented [18].

The use of agile and flexible manufacturing processes is another aspect of resilient manufacturing in smart manufacturing systems. The use of modular, flexible production systems that can quickly adapt to changes in demand or production requirements is referred to as agile manufacturing. Meanwhile, flexible manufacturing makes use of digital technologies to allow for the rapid reconfiguration of production lines and processes. Implementing these approaches enables manufacturing systems to respond quickly to unexpected events and changes in market conditions, thereby increasing their resilience [19].

To summarise, Smart Manufacturing (SM) is a term used to describe the integration of communication and computing technologies in industrial settings to achieve intelligent coordination across the supply chain, enterprise, and shop floor levels. This concept has become increasingly relevant in recent years with the emergence of Industry 4.0. SM aims to improve occupational health and safety, reduce resource and energy consumption, incorporate robots into manufacturing processes, and integrate the virtual world into manufacturing. The core components of SM include smart sensors and processors, the Industrial Internet of Things (IIoT), data analytics, cloud computing, material informatics, zero-waste manufacturing, 3D printing/additive manufacturing, cyber security, and international standards. SM is a critical technology for achieving sustainable and resilient manufacturing practices. By reducing waste and energy consumption, digital manufacturing can help to achieve sustainability goals. Additionally, digital technologies can aid in the improvement of supplier relationships, leading to more sustainable practices throughout the supply chain. Resilient manufacturing is also essential for SM systems to withstand disruptions and recover quickly from unexpected events such as natural disasters or supply chain disruptions [12, 13, 14, 17, 18, 19].

2.1.1 SMs in SMEs

Small and medium-sized enterprises (SMEs) play a significant role in the global economy, helping to create jobs and boost economic growth. SMEs typically employ fewer than 250 people and generate less than €50 million in annual revenue. Manufacturing industries contribute significantly to SMEs, with many businesses relying on traditional manufacturing methods. Adoption of digital manufacturing technologies, on the other hand, can boost SMEs' competitiveness and productivity [20].

Digital manufacturing, also known as Industry 4.0 or smart manufacturing, refers to the use of digital technologies to automate and optimize manufacturing processes. Digital manufacturing technologies include robotics, artificial intelligence, the

Internet of Things (IoT), and data analytics. These technologies can improve production efficiency, reduce waste and defects, and enable businesses to respond quickly to changes in demand. In recent years, SMEs have increasingly embraced digital manufacturing to remain competitive in a rapidly changing global market [21].

The adoption of digital manufacturing technologies in SMEs varies significantly across regions. In Europe, where SMEs play a crucial role in the economy, there has been a significant push to promote the adoption of digital technologies in manufacturing. According to a 2020 report by the European Commission, 50% of SMEs in the European Union (EU) have adopted innovative activities and actively using advanced manufacturing technology. The report found that the most adopted technologies were 3D printing, collaborative robots, and advanced sensors. However, the adoption rate varied significantly by country, with the highest rates observed in Sweden, Denmark, and the Netherlands [22].

One of the main challenges facing SMEs in the adoption of digital manufacturing technologies is the lack of resources and expertise. SMEs often lack the financial resources to invest in expensive technologies or the in-house expertise to implement and manage these technologies. Furthermore, many SMEs are reluctant to invest in modern technologies due to the perceived risks and uncertainty associated with the adoption process [23].

To overcome these challenges, governments and industry associations have implemented various initiatives to support SMEs in their adoption of digital manufacturing. For example, the European Union has established the European Digital Innovation Hubs (EDIHs) to provide SMEs with access to expertise, technology, and funding to support their digital transformation. Furthermore, the European Commission has launched the Digital Innovation Hubs (DIHs) initiative to help SMEs access digital manufacturing technologies and services [24].

SMEs play a vital role in the global economy, and the adoption of digital manufacturing technologies can enhance their competitiveness and improve their productivity. While the adoption of these technologies varies significantly across regions, there has been a significant push in Europe to promote the adoption of digital manufacturing in SMEs. However, SMEs still face significant challenges in the adoption process, including a lack of resources and expertise. Governments and industry associations have implemented various initiatives to support SMEs in their adoption of digital manufacturing, but more needs to be done to ensure that SMEs can fully benefit from these technologies [25].

2.2 Production metrics in SMEs

Medium-sized enterprises (SMEs) deal with significant obstacles in optimizing their production processes to fulfil consumer needs, enhance efficiency, and ensure overall company success in today's highly competitive global market. To accomplish these objectives, SMEs must employ effective production measures that allow them to evaluate and improve their operational efficiency. This section will look at five primary production metrics - Availability, Performance, Quality, Overall Equipment Effectiveness (OEE), and Productivity - and how they interact to improve manufacturing performance for SMEs.

Availability: Availability is a critical matrix that measures the operational uptime of manufacturing equipment in digital manufacturing. It is calculated by dividing the actual Run Time of the machine by the Planned Production Time, which represents the duration when the machine was scheduled to be in an active state. Monitoring Availability allows manufacturers to identify and address downtime issues promptly, improving overall equipment efficiency [26].

$$Availability = Run\,Time \,/\, Planned\,Production\,Time \tag{2.1}$$

Performance: Performance is another crucial metric in digital manufacturing, focusing on evaluating the efficiency of production processes. It is calculated by multiplying the Ideal Cycle Time (defined by the user) with the Total Count of products produced, divided by the Run Time. By monitoring and optimizing Performance, manufacturers can identify bottlenecks, optimize cycle times, and enhance productivity [26].

$$Performance = (Ideal\,Cycle\,Time \times Total\,Count) \,/\, Run\,Time \tag{2.2}$$

Quality: Quality is a fundamental metric digital manufacturing, representing the ratio of Good Count (products without faults) to the Total Count of products produced. By monitoring and improving Quality, manufacturers can ensure customer satisfaction, reduce waste, and minimize the costs associated with rework or product recalls [26].

$$Quality = Good\ Count\ /\ Total\ Count \tag{2.3}$$

Overall Equipment Efficiency (OEE): Overall Equipment Efficiency (OEE) is a comprehensive metric that combines Availability, Performance, and Quality to provide a holistic measure of manufacturing productivity. By analysing OEE, manufacturers can identify the root causes of inefficiencies, track performance trends over time, and implement targeted improvement strategies to optimize overall productivity and competitiveness [26].

$$OEE = Availability \times Performance \times Quality \text{ Or} \tag{2.4}$$

$$OEE = (Good\ Count \times Ideal\ Cycle\ Time)\ /\ Planned\ Production\ Time \tag{2.5}$$

Productivity: Productivity is a significant metric in digital manufacturing that focuses on the output achieved with Availability and Performance, assuming a constant level of Quality. It measures the efficiency of converting available resources and operational time into tangible output. Productivity can be calculated by dividing the Total Count of good products by the Planned Production Time. By monitoring and optimizing productivity, manufacturers can assess the effectiveness of their operations, identify areas for improvement, and make informed decisions to increase output and overall efficiency [26].

$$Productivity = Results/working\ Time \tag{2.6}$$

2.3 Artificial Intelligence in Smart Manufacturing System

Smart Manufacturing is a modern approach to manufacturing that incorporates advanced technologies to optimize production processes, increase efficiency, and reduce costs. One of the key technologies that have revolutionized Smart Manufacturing is Artificial Intelligence (AI). AI has emerged as a powerful tool for managing and optimizing manufacturing processes, and it has enabled manufacturers to realize significant improvements in production efficiency, quality, and safety [27].

AI encompasses several technologies that are being used in manufacturing, including machine learning, deep learning, natural language processing, computer vision, and robotics. Machine learning algorithms can analyse data from sensors and other sources to detect patterns and predict equipment failures, while deep learning algorithms can identify complex patterns in production data that may not be apparent to human analysts. Natural language processing can be used to interpret textual data, such as maintenance manuals, to extract insights that can be used to optimize maintenance schedules. Computer vision can be used to analyse video data from cameras to detect quality issues or unsafe conditions, while robotics can automate manual processes to improve efficiency and reduce costs [28].

AI has the potential to transform every aspect of Smart Manufacturing, from production planning and scheduling to quality control and maintenance. AI-powered systems can monitor and analyse vast amounts of data in real time, enabling manufacturers to detect and respond to issues quickly and efficiently. For example, AI algorithms can analyse sensor data to detect anomalies in machine performance, predict equipment failures, and recommend preventive maintenance measures [29].

In addition to predictive maintenance, AI can also improve product quality by identifying patterns and trends in production data. By analysing data from sensors, cameras, and other sources, AI can detect defects and deviations from expected production norms. This enables manufacturers to take corrective actions quickly, reducing the risk of product recalls and improving customer satisfaction [30].

Furthermore, AI can optimize production processes by identifying opportunities for process improvements and automation. By analysing production data, AI can identify bottlenecks, inefficiencies, and areas where manual processes can be automated. This enables manufacturers to improve production efficiency, reduce waste, and improve overall profitability [12].

However, the adoption of AI in Smart Manufacturing is not without its challenges. One of the key challenges is the need for data integration and standardization. Smart Manufacturing systems generate vast amounts of data from multiple sources and integrating and standardizing this data is critical to achieving the full potential of AI. Furthermore, there is a need for skilled personnel who can design, implement, and manage AI-powered Smart Manufacturing systems [31].

AI has emerged as a powerful tool for managing and optimizing smart manufacturing processes. It has the potential to transform every aspect of manufacturing, from production planning and scheduling to quality control and maintenance. While there are challenges to the adoption of AI in smart manufacturing, the potential benefits are significant, and AI is likely to play a critical role in the future of manufacturing [32].

Modern manufacturing plants employ powerful data acquisition systems to electronically collect and transfer data from all the organization's processes. Many manufacturing variables are continuously measured at various stages and their values are stored in databases. This data may be related to product characteristics, machine characteristics, production line (i.e., which machine has been used with which setup parameters), human resources that operate the production line (i.e., worker experience level, shift type), raw materials used in the process, the environment (moistness, temperature, etc.), sensors attached to the machines (vibration, force, pressure, tension, etc.), machine failures / maintenances, product quality, etc [33].

In the era of big data, data analytics and machine learning are being applied to a wider range of areas in process industries. Figure 2.2 illustrates the penetration of these methods into various hierarchies in process industries, including passive applications in low-level control loops such as process monitoring and soft sensing, as well as active applications such as optimal control and high-level decision-making. The passive applications aim to help industrial practitioners better observe and manipulate the process and identify important variations without directly influencing the processes. In contrast, active applications result in decisions that directly impact industrial processes [34].

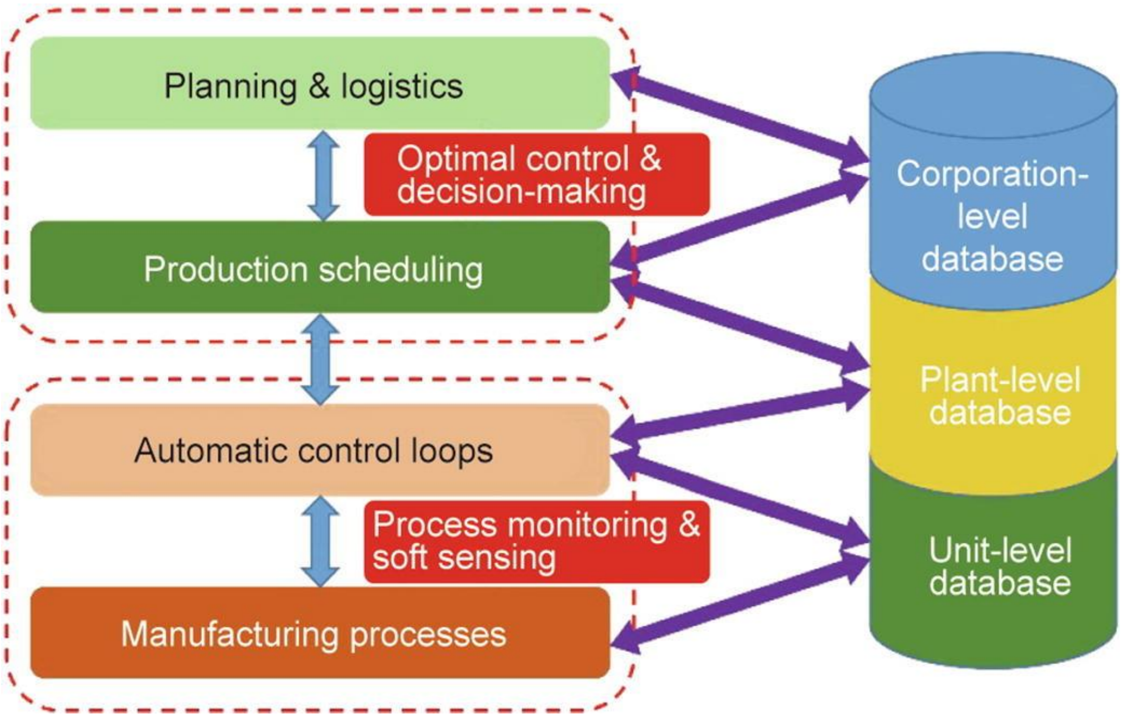

Figure 2.3.1 Applications of machine learning and data analytics in process industries [34]

Furthermore, data analytics in the industrial sector may be utilized to accelerate development in the following areas [35]:

- ➢ Utilize data analytics to increase assembly line productivity.
- ➢ Enhanced customer experience, including personalized value offerings and locating them.
- ➢ Inventory control by
- Real-time knowledge of inventory and visibility of supply lines
- Delivery route improvements
- ➢ Reduce loss brought on by late, damaged, or missing items while being transported, and provide real-time asset management by using real-time warnings.
- ➢ Improve the quality and packaging of the product while minimizing mistakes and adjustments made during product creation by,

- Simulations supported by analytics
- Product simulation

➢ Predictive maintenance extends the useful life of assets by,

- Asset administration.
- Raising the availability of assets.
- The detection of errors and flaws.
- Stopping unnecessary downtimes.

➢ Increase supply chain visibility with the use of location-based IoT technologies and actionable data.

The above areas widely use data analytics to accomplish different tasks. Using the data to teach the machine to learn on its which in turn can predict various outputs depending upon the input is the way to move forward from data analytics to machine learning. The section below describes the machine learning methods and followed by their applications in the manufacturing domain.

2.4 Machine Learning

Machine learning is a subset of Artificial Intelligence (AI) that focuses on the development of computer algorithms which can learn from the data feed by users without explicitly being programmed. In other words, ML is the process of using statistical techniques to enable machines to improve their performance on a task by learning from data. Data being the key factor across the ML domain, generating, capturing, and storing the huge amount of data and ease of creating machine learning algorithms has been making ML immensely popular in different industries. Arthur Samuel of IBM, designed the first computer program that could learn as it ran, allowing it to defeat a human in checkers in 1952 [36, 37, 38]. This was an important turning point in ML since a machine had never previously defeated a person in a game; it was also the first time the term 'ML' was stated [38]. Samuel in 1959 define ML as "Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort [39]".

There are several definitions of machine learning, but one of the most widely accepted is from Tom Mitchell, a computer scientist and former department head at

Carnegie Mellon University. Mitchell defined machine learning as "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E [40]”.

ML methods read data points created within a certain application area. A single data point is distinguished by several attributes. We find it useful to categorize data point attributes into two categories: features and labels. Features are qualities that we can simply measure or compute in an automated manner. Labels are qualities that cannot be easily quantified and frequently signify some higher-level truth (or amount of interest) whose discovery often necessitates the assistance of human specialists. ML seeks to learn to predict (approximately or estimate) the label of a data point based purely on the attributes of that data point. Formally, the prediction is generated as the function value of a hypothesis map, with the characteristics of a data point serving as the input argument. Any ML method can only consider a portion of all potential hypothesis mappings since it must be implemented with restricted processing resources. This portion of the hypothesis is called hypothesis space or the model underlying an ML model. Not only for predicting the hypothesis, but Machine learning is also widely used to understand the relation between the input and the output data. The components of Machine Learning can be broken down into three components: Data, Model and Loss functions [36]. These components are broken down into subtopics further in this paper.

There are four primary types of machine learning: supervised, unsupervised, semi-supervised and reinforcement learning, based on how ML approaches evaluate the validity of various hypothesis maps. Despite the three, there are new and advanced types of machine learning models are being introduced with the increasing computation power and the data being generated by smart [41].

2.4.1 Supervised learning

Supervised ML methods are the focus of this research. Supervised learning uses a training dataset with labelled data points. Data points are called labelled if the label values are known. This machine learning method focuses to learn an approximate predictor $h(x)$ that maps inputs x to output y with the help of annotation ($x1$,

$y1$), ($x2$, $y2$),...,(xN , yN)[41, 42]. In supervised learning, the algorithm examines a labelled dataset to produce an inferred function that may be used on samples that have not yet been seen. Output variables may be either categorical or continuous. In the first case, the problem is known as a classification task, and a classic example would be generating a model to detect process failures or predict the quality level of new production batches from a dataset containing the physical properties x and the quality level y of completed production batches. In contrast, if the variables are continuous, the issue is known as a regression job, and an industrial example may be predicting the thickness or surface roughness of objects processed by a numerical control system. This can be done using image recognition using computer vision where the image data is turned to a vector of features to generate variable x [43]. The illustration below describes the general algorithm of most of the supervised learning models [43].

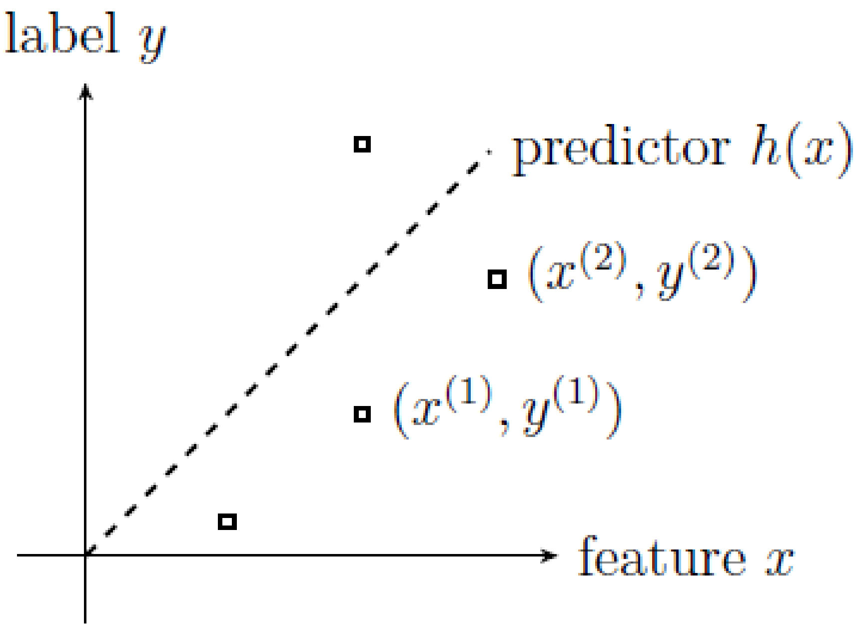

Figure 2.4.1.1 Supervise ML curve [42]

As this research focuses on using supervised machine learning methods to understand the relationship between process variability (availability, performance) on productivity in manufacturing systems, some of the supervised machine learning methods that are used are described below:

Regression is a type of analysis that aims to uncover connections between distinct factors. It focuses on understanding the relationship between a single outcome (known as the dependent variable) and one or more explanatory factors (known as independent variables). These factors can be things like measurements, observations, or characteristics [42, 44].

In linear regression, we use a simple equation to represent this relationship. The equation consists of a constant term (B0) and coefficients (B1, B2, and so on) multiplied by each independent variable (X1, X2, and so on). We add them up to predict the value of the dependent variable (Y). The letter 'e' represents the residual or error, which accounts for any unexplained variation [44].

$$Y = B0 + B1X1 + B2X2 + ... + e \qquad (2.3.1.1)$$

Linear regression helps us make predictions about continuous outcomes. For instance, we can estimate the price of a house based on factors like its size, location, and number of bedrooms. It is important to note that linear regression is different from classification, which deals with predicting labels from a limited set of options [44].

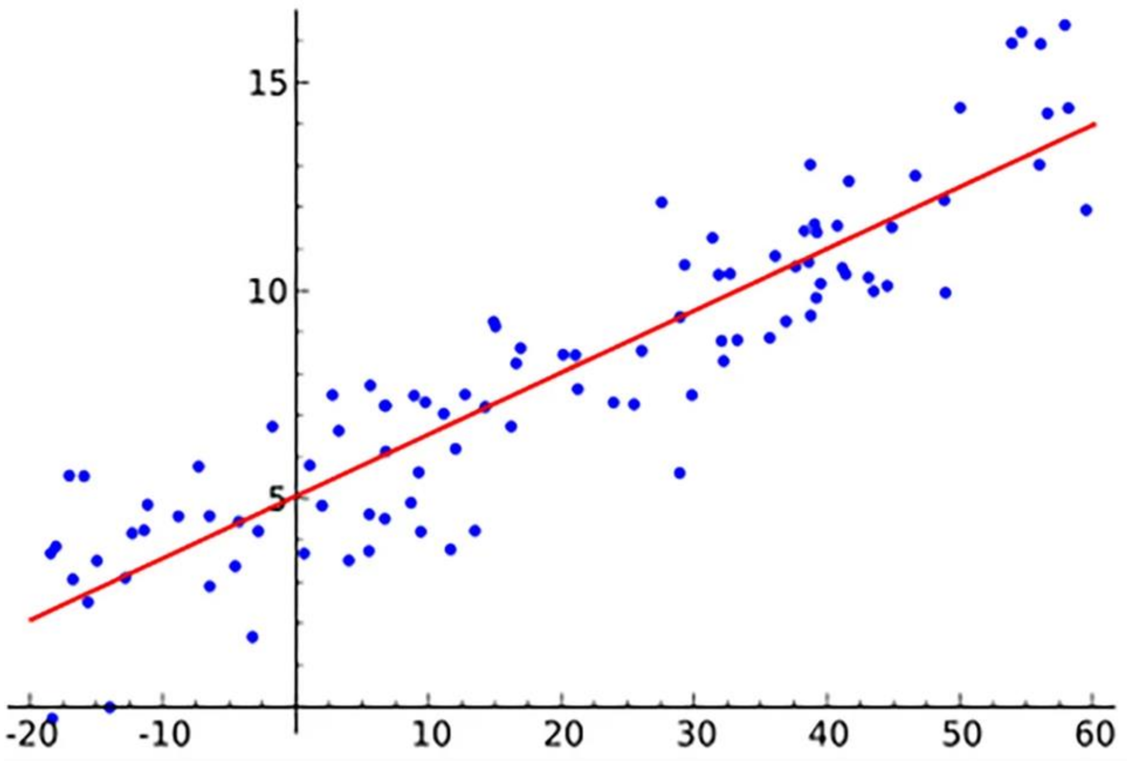

Figure 2.4.2.2 Sample representation of linear regression model [45]

Decision trees are a type of analysis that organizes attributes based on their values. They are primarily used for classification purposes. A decision tree is composed of nodes and branches. Each node represents a group of attributes to classify, and each branch represents a value for that node [44].

In data mining and machine learning, decision tree learning is a technique that uses a decision tree as a model to predict outcomes or target values based on observations. These tree models can also be called classification trees or regression trees, depending on the context. To improve the performance of decision tree classifiers, post-pruning techniques are often employed. These techniques evaluate the tree's performance and remove unnecessary nodes based on a separate validation set. Any node can be deleted and assigned the most common class of training instances within that node [44].

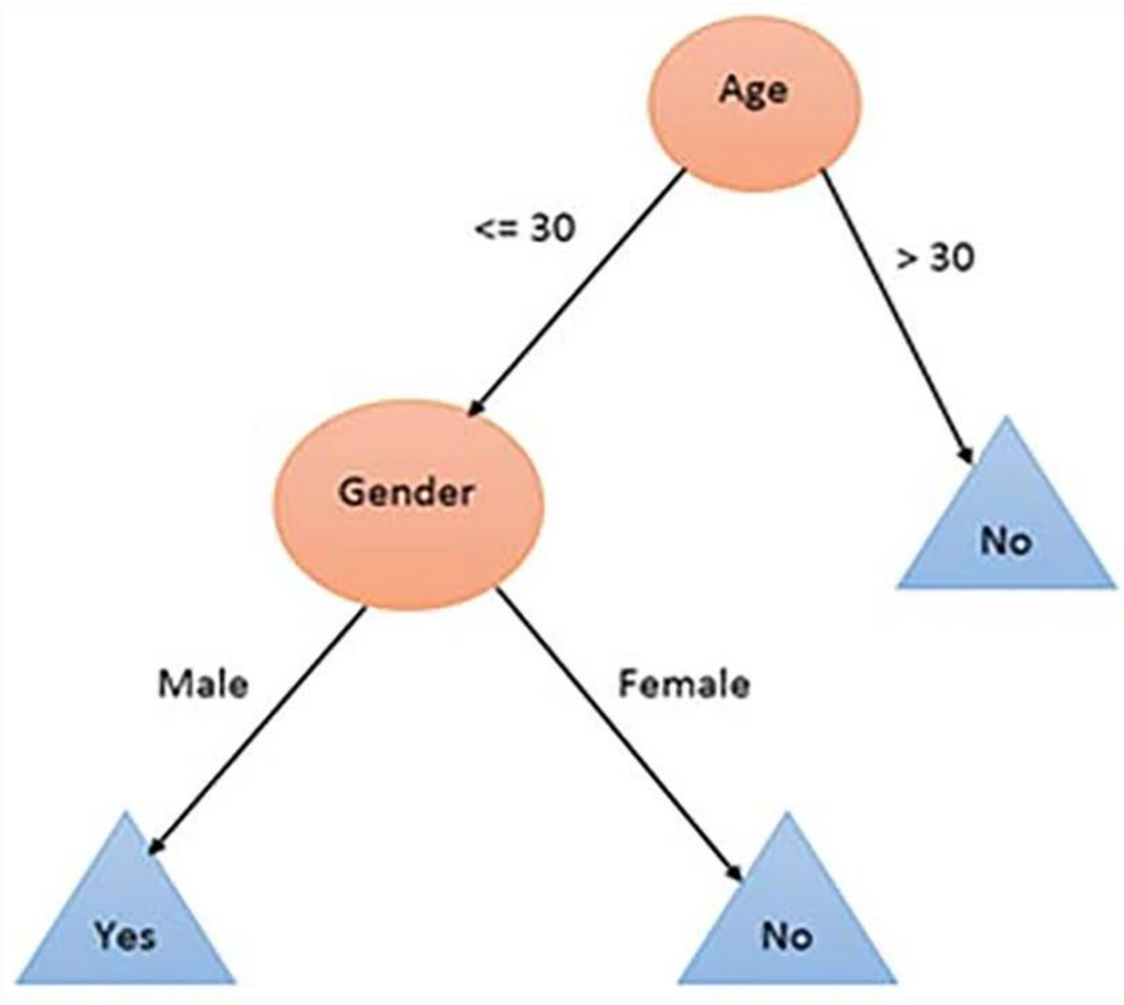

Figure 2.4.3 Decision Tree Map [44]

Random forest is a fancy term for a group of decision trees working together. In a random forest, each tree gets to vote on how to classify a new object based on its attributes. The forest then tallies up the votes from all the trees and picks the class with the most votes as the final decision [42, 44].

Now, let us understand how each tree in the forest is grown [44]:

- Start by randomly picking a sample of cases from the training set. This sample will be used to teach each tree how to make decisions.

- Next, a small number of input variables is randomly selected at each node of the tree. This helps in deciding how to split the node and make decisions. The same number of variables is used consistently as the entire forest grows.

- Each tree in the forest is allowed to grow as much as possible. There are no limits on the size or complexity of the trees. The goal is to make them as powerful as they can be.

By using this collection of decision trees, a classifier can be created that is used for both classification and regression tasks. It provides information about accuracy and the importance of different variables in making predictions. This way, one can see how reliable the results are and which factors matter the most [41, 43, 44].

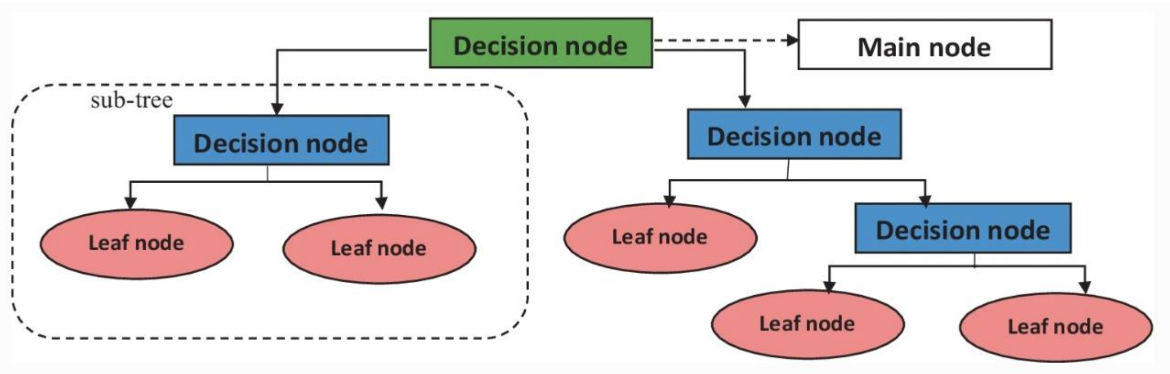

Figure 2.4.4 Random Forest Structure [44]

Neural Networks use artificial neurons to communicate and process information inspired by how our brains work. A typical neural network, like ANNs, has different layers: an input layer, a hidden layer, and an output layer. There are distinct types of ANNs, such as CNNs, RNNs, and Deep Belief Networks [41, 42].

CNNs are great at analysing matrix-like data, making them useful for image processing tasks. In manufacturing, CNNs can be used for image-based quality control or process monitoring. By converting sensor data into 2D images, CNNs can also help detect and diagnose machine failures [41, 42].

RNNs are designed to handle sequential input data, like time series or sequential images. In manufacturing, RNNs are ideal for analysing sensor data or live images from machines and processes. They can predict real-time performance, the remaining useful life of machinery, process behaviours, or production indicators for scheduling [41, 42].

Neural networks, or NNs, are powerful systems made up of interconnected neurons. These networks can handle regression and classification tasks simultaneously, although each network typically focuses on one task. The network's behaviours are determined by the weights assigned to each input connection. Initially, these weights are set randomly, and then the network is trained using a training set. During training, the network adjusts its weights to make its output values closer to the desired outputs for each instance in the training set. This iterative process helps the network learn and improve its predictions [41, 42].

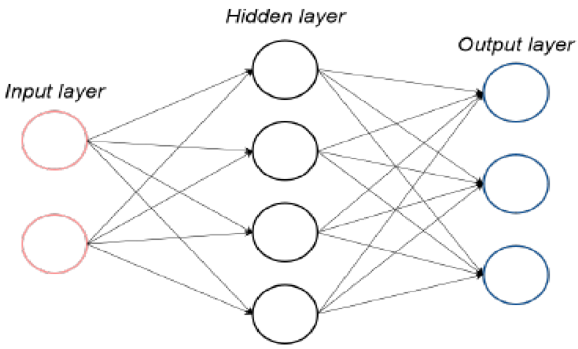

Figure 2.4.5 Sample of Artificial Neural Network schema with one hidden layer [41]

2.4.2 Unsupervised learning

Unsupervised Learning is all about working with datasets that do not have any labels or known answers. The main aim here is not to make predictions but to discover and identify patterns in the data. These patterns might be unknown or only partially understood. That is why Unsupervised Learning is sometimes called descriptive learning—it helps us uncover hidden knowledge [41, 43].

When it comes to Unsupervised Learning, we can broadly divide it into three sub-areas: clustering, density estimation, and dimensionality reduction. Clustering involves grouping objects based on their similarities. For example, in marketing, we might want to find groups of customers who have similar purchasing behaviours. Algorithms like Hierarchical Clustering or K-Means are often used for this purpose when we do not have information about group membership [41, 43].

Density estimation is a set of techniques used to discover useful properties or estimate the underlying probability distribution of a dataset. For example, we might want to determine if our data is skewed or has multiple peaks. Basic approaches like rescaled histograms can be used, but more advanced techniques like Parzen Windows and vector quantization are also available [41, 42, 43].

Dimensionality reduction is frequently used, especially when dealing with Big Data, to compress data without losing essential information. The goal is to capture the essence of the data while reducing its size. Principal Component Analysis is a commonly used technique for this task, but neural network models like Autoencoders can also be employed to find the best-compressed representation of the original data. All Deep Learning models are designed to capture hidden representations and meaningful relationships within the data. That is why Deep Learning is sometimes referred to as Representational Learning [43].

2.4.3 Semi-supervised learning

Semi-Supervised Learning provides a way to benefit from both supervised and unsupervised learning methods, offering a balance between accuracy and labelling costs. In traditional unsupervised learning, there is no input guidance during training, which reduces labelling efforts but often leads to less accurate results. To address this, researchers have turned to data augmentation techniques. By augmenting existing datasets with additional inputs and labels generated in a controlled manner, using transformations like rotation, translation, flipping, and noise injection, the dataset size can be increased without incurring extra labelling costs [40, 41, 42, 43].

Adversarial data augmentation is another approach used in semi-supervised learning. It involves generating synthetic datasets using generative models like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). These models can create new images for training machine learning models, expanding the dataset while keeping costs low. However, the benefits obtained from data augmentation alone have limitations, as real data tends to be more valuable than synthetic data [42].

Therefore, researchers have increasingly focused on the combination of supervised and unsupervised learning, known as semi-supervised learning. This approach leverages both labelled and unlabelled data during training. Semi-supervised learning methods can be categorized into two main groups: data augmentation-based methods and semi-supervised mechanism-based methods [43].

Data augmentation-based methods involve enriching the dataset using various techniques to create additional labelled examples. These methods exploit the relationships and patterns within the data to enhance the learning process. On the other hand, semi-supervised mechanism-based methods utilize specific algorithms or mechanisms designed to effectively utilize both labelled and unlabelled data. These methods aim to leverage the unlabelled data to improve the model's performance and generalization abilities [41].

By combining supervised learning with unsupervised learning in semi-supervised learning, machine learning models can benefit from the strengths of both approaches. This allows for more accurate predictions while reducing the costs associated with labelling substantial amounts of data. The field of semi-supervised learning continues to evolve, offering promising opportunities for enhancing machine-learning models in various domains [42].

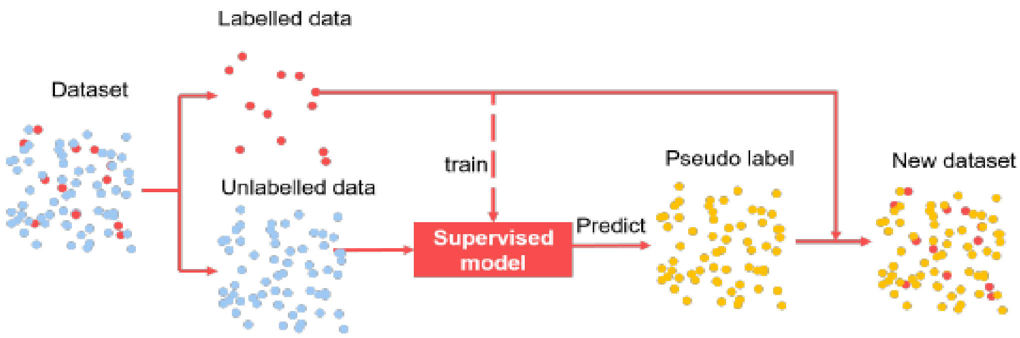

Figure 2.4.6 Data augmentation-based method [42]

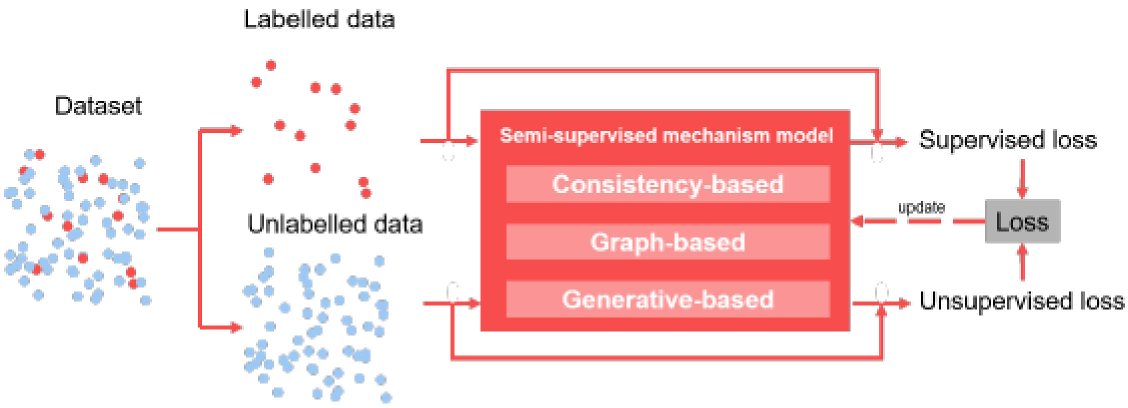

Figure 2.4.7 Semi-supervised based method [42]

2.4.4 Reinforcement learning

Reinforcement Learning stands out from other machine learning approaches because it learns through interactions with an environment. Instead of mapping inputs to outputs, it creates a map of situations to actions. Just like how humans learn, Reinforcement Learning does not rely on pre-existing datasets. Instead, it uses agents that learn by doing, following a trial-and-error approach with rewards. The agent can interact with the environment, taking predefined actions based on a predetermined policy. Each action changes the state of the system and receives a reward signal. The agent's goal is to maximize the total reward, learning the best actions for different scenarios or states. Q-learning is a popular algorithm in Reinforcement Learning that learns action values without needing an explicit model of the environment [40,42].

The learning process in Reinforcement Learning can also be supported by supervised and/or unsupervised algorithms to optimize the exploration and exploitation of the agent's action space. When these algorithms are based on neural networks, it is called Deep Reinforcement Learning. Double Q-Learning is an example of using Deep Learning models to improve the classic Q-learning algorithm [42,43].

Regardless of the specific implementation, the goal of a Reinforcement Learning algorithm is to create an artificial agent capable of making good decisions based on the current state of the environment and its past experiences. For example, in industrial settings, RL agents could be used to automate ordering strategies in multi-tier supply chain networks or adjust production parameters to maximize yield while minimizing operating costs [41].

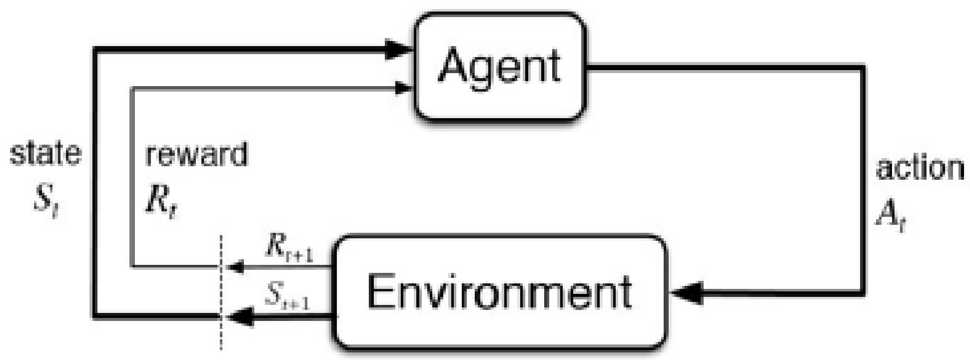

Figure 2.4.8 Overview of Reinforcement Learning [41]

2.5 ML in the manufacturing domain

The field of machine learning (ML) holds great promise for revolutionizing the manufacturing industry. Many studies have demonstrated the impressive performance of ML techniques in various manufacturing applications [35, 38, 40]. However, despite these advancements, the practical implementation of ML solutions remains a daunting task, especially for non-experts in the industry. One of the major challenges lies in formulating the specific problems that ML can address effectively [41, 42, 44].

Tingting et. al [43] address this challenge by introducing the concepts of Four-Know and Four-Level, which provide a framework for formulating ML tasks in manufacturing. Furthermore, they explore the benefits of applying ML in the manufacturing domain, highlighting ML use cases categorized using the Four-Know and Four-Level concepts. Additionally, they provide an overview of current trends in ML studies that have been formulated using the Four-Know and Four-Level framework [42].

Formulating ML tasks in manufacturing is the first crucial step towards leveraging the power of ML techniques. It requires a deep understanding of the manufacturing processes, machines, and systems, along with the ability to translate real-world problems into ML problems. By categorizing ML tasks using the Four-Know and Four-Level concepts, can gain valuable insights into how ML can contribute to several aspects of manufacturing optimization [42].

The Four-Know categories, namely Know-what, Know-why, Know-when, and Know-how, provide a structured approach to defining the objectives of ML tasks. Know-what focuses on understanding the current states of machines, processes, or production systems. Know-why delves into uncovering patterns and causal relationships behind events. Know-when enables timely predictions and forecasting of key variables. Lastly, Know-how offers recommendations for decision-making and adaptive strategies [42].

In addition to the Four-Know framework, the Four-Level concept recognizes that ML can be applied at diverse levels of manufacturing, including the product, process, machine, and system levels. Each level presents unique challenges and opportunities for applying ML techniques. By considering the Four-Level perspective, can identify ML use cases that span the entire manufacturing ecosystem [42].

By exploring ML use cases categorized using the Four-Know and Four-Level concepts, we gain a comprehensive understanding of the benefits of applying ML in manufacturing. These use cases cover a wide range of applications, including quality control, predictive maintenance, process optimization, and inventory management. Understanding these benefits can inspire and guide non-experts in the manufacturing industry to start developing ML solutions for their specific problems [42].

This section aims to address the challenges faced by non-experts in implementing ML solutions in manufacturing. Introducing the Four-Know and Four-Level concepts provides a structured approach to formulating ML tasks and highlights the benefits of applying ML in various manufacturing domains. An overview of contemporary trends ensures that readers are up to date with the latest advancements in ML studies formulated using the Four-Know and Four-Level frameworks [42].

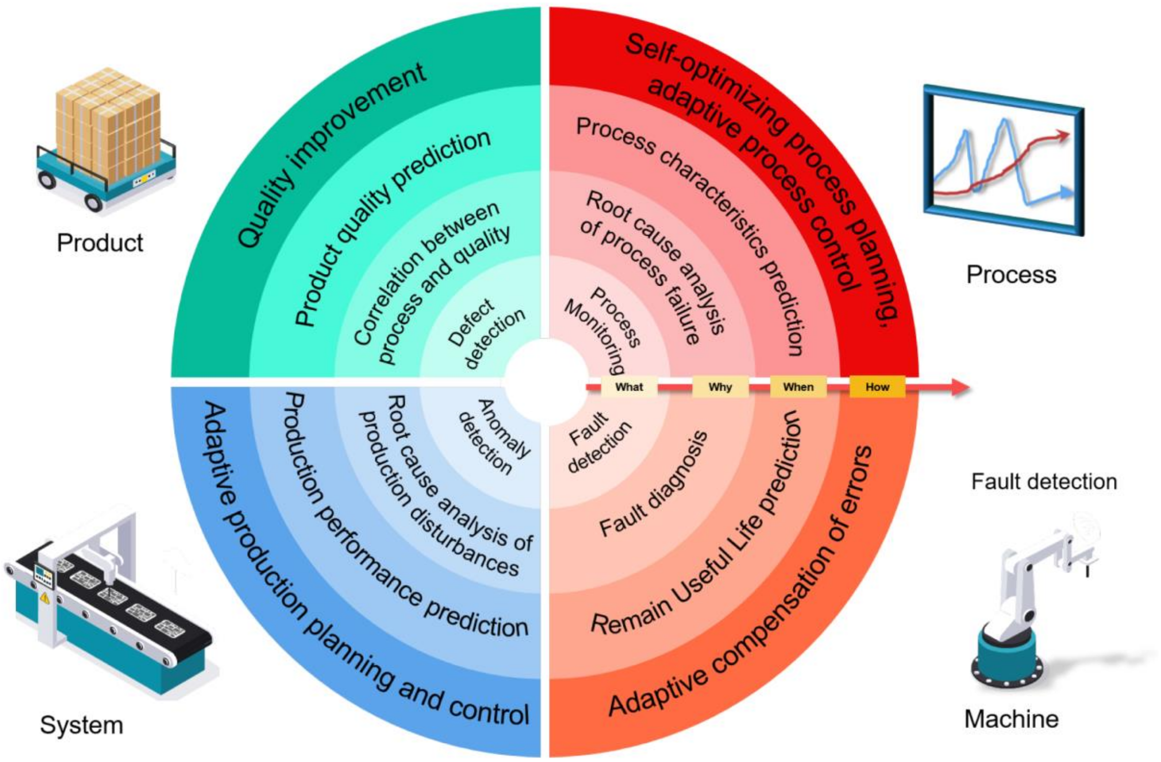

Figure 2.5.1 Categorization of ML application in manufacturing in Four-level and Four-know [42]

3. METHODOLOGY

This section provides a brief overview of the method used to answer the research questions that were formulated. The overall method is mostly focused on the pipeline of applying machine learning in manufacturing. The background is provided in section 2 of the research, methodology will follow the literature review and provide a solid background for the results and analysis section. This section will display the methods used in this research.

In recent years, an increasing number of manufacturing industries have started using machine learning (ML) to their advantage by creating ML solutions for various industrial fields. However, applying ML to real-world problems comes with its challenges, especially for small and medium-sized enterprises (SMEs). These businesses often struggle to develop their own ML solutions because the existing commercial options are usually confidential and hard to access. Therefore, this section aims to provide a beginner-friendly guide for applying ML from scratch. The process of implementing machine learning in manufacturing can be broken down into six simple steps: (I) collecting data, (ii) cleaning the data, (iii) transforming the data, (iv) training the model, (v) analysing the model, and (vi) putting the model into action, as shown in Figure [42].

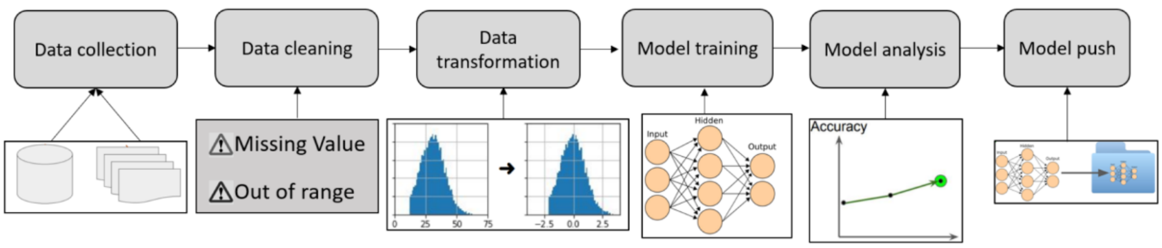

Figure 2.5.1 Generic method of applying machine learning in manufacturing [42]

3.1 Data collection

Data being an atom of the Machine learning process, it was critical to get the data that would solve the research question. There were a lot of open-source data that were accessible through different platforms like Kaggle, Google, GitHub, etc. But the thesis aims to use the data from SMEs and prepare it for the model training and test. The data was requested from the university and the researcher from the university provided the dataset. It was requested to keep the confidentiality of the source of the data and the management system. Thus, it was agreed to provide the data in Tabular format using Microsoft Excel. The data that was collected is from a machine used by an SME where different researchers from the university were working in collaboration. The data is time series data collected in everyday basics for 3 months period.

The data that were calculated are described in subsection 2.1.2. The basis of the calculation was the measured data by sensors which are described below:

- Run time –the time when work is active, measured by a sensor
- Planned Production Time – the time when the machine was in the state on
- Ideal Cycle Time – defined by the user
- Total count - measured by a sensor
- Good count – products without fouls
- Total count – total products

3.2 Data cleaning

After the required data is collected, the next process is to clean the data. This process involved removing null values, checking duplicated values, missing values, and removing them. The data shows that there were days when there was no production thus the data collected has a value of 0, which was removed from the dataset. The process was iterative and after this process, a clean dataset was achieved for further processing.

3.3 Data transformation

Data transformation for this thesis is characterized by feature selection and pre-processing. Feature selection and pre-processing are vital steps in developing accurate and efficient predictive models [45]. In the given dataset, the initial feature set comprised Date, Workstation, User, and Avail. %, Perf. %, Quality, OEE, Productivity, Op. active, off time, Short Stop, Long Stop, Working time, and Total time. However, since Quality was consistently at 100% in the dataset, it did not provide any discriminatory information for the output variable. Therefore, we excluded Quality from the final feature set and its relation to OEE was not considered.

To pre-process the data, the focus was on selecting key features that have a significant impact on the output variable, which in this case is productivity. Consequently, we chose Avail. % And Perf. % As the input features as they relate to the availability and performance of the workstation. Normalization was done by converting the percentage values to decimal values. For example, if we had a value of 70%, we would represent it as 0.70 after normalization. This conversion ensured that the features, specifically Avail. % And Perf. %, were brought to a common scale for accurate modelling and analysis.

By converting the percentage values to decimals, we achieved consistency and prevented any potential bias that could arise from differences in magnitude. This normalization process allows the model to interpret and compare the features effectively, as they are now represented on a standardized scale between 0 and 1 [45].

Normalization is an essential step in pre-processing data for machine learning tasks. It ensures that features with different units or scales are brought to a similar range, preventing certain features from dominating others simply due to their larger numerical values [45]. In this case, converting Avail. % And Perf. % From percentages to decimals facilitated fair and meaningful comparisons between these features during model training and prediction.

By normalizing the data in this manner, we create a consistent foundation for accurate analysis and modelling, enabling the machine learning algorithms to effectively learn from and interpret the patterns and relationships within the dataset. This normalization process enables us to bring these features to a common scale, mitigating any potential bias stemming from differences in magnitude [45].

Feature selection plays a crucial role by eliminating irrelevant or redundant features, thereby reducing the dimensionality of the problem, and enhancing model efficiency. By carefully selecting Avail. % And Perf. % As input features, we aim to capture the essential factors that contribute to productivity. These features provide valuable insights into the availability and performance of the workstation, allowing the model to learn and make predictions based on pertinent information [45]

Although OEE was initially considered as a feature, it was not included in the final selection due to the constant 100% quality value. Nevertheless, OEE can be indirectly related to productivity through its components, such as availability and performance. By utilizing Avail. % And Perf. % As input features, we can still capture the essential elements of OEE that contribute to productivity, even if the direct OEE value itself was not used in the model.

3.4 Model Selection and Training

Once the data transformation has been done, the next step is to train the machine learning model. Although many machine learning models can be trained to answer the research question, it is considered that the data were collected from SMEs and the limitation factors like resources, budget, and data management were considered. Thus, four of the traditional machine learning models were selected to be trained. Those models are Regression (non-linear), Decision Tree, Random Tree, and Artificial Neural Network (ANN).

Several parameters are considered throughout the model selection process to choose the best algorithm for the task. MATLAB, a widely used programming language and environment for scientific computing, was used as the platform for training the models in this scenario. MATLAB is an excellent choice for this project since it provides a complete range of tools and algorithms particularly built for machine learning and data analysis activities. Its extensive algorithm library and a user-friendly interface allow for the quick building, training, and assessment of machine-learning models [46].

MATLAB has broad support for regression analysis and classification problems, allowing for the construction of a wide range of techniques. Furthermore, MATLAB's visualization tools enhance data exploration and comprehension, which aids in model selection. Because of the platform's versatility, researchers and engineers can build and fine-tune their models, optimizing them for specific goals and datasets [46].

To train all the models, the 80-20 rule was used. Eighty per cent of the data was used to train the model and twenty per cent was used to test the model.

3.5 Model Analysis and Evaluation

After training the four chosen models – non-linear regression, Decision Tree, Random Forest, and Artificial Neural Network (ANN) – it is time to examine and evaluate their performance in predicting the required output variable.

Various evaluation metrics can be used to assess the performance of the models. Root Mean Squared Error (RMSE), Mean Square Error (MSE), and R-squared (R^2) score are three often used metrics in the context of regression tasks. These metrics give information on the accuracy, precision, and goodness of fit of the models [47].

The average divergence between expected and actual values is measured by RMSE. It computes the square root of the average of the squared discrepancies between the expected and actual values to get the overall prediction error. A lower RMSE number implies that the model is better fitted to the data [47].

$$RMSE = \sqrt{\Sigma \quad {}^{n}{}_{i=1} (X - Y)} \quad {}^{2}/N \qquad (3.4.1)$$

where X = actual value, Y is the predicate value and N is the number of data points

MAE represents the average magnitude of errors. It computes the average absolute difference between anticipated and observed values. A lower MAE number, like a lower RMSE value, suggests a more accurate model [47].

$$MSE = \sqrt{\Sigma \quad {}^{n}{}_{i=1} (Y - X)/N} \tag{3.4.2}$$

where X = actual value, Y is the predicate value and N is the number of data points

The R-squared (R^2) score is a statistical metric that shows how much of the variance in the dependent variable can be explained by the independent variables. It has a value between 0 and 1, with one signifying a perfect match. The R^2 value indicates how well the model matches the data [47].

$$R^2 = 1 - \frac{\Sigma(X - Y)}{\Sigma(X - Z)} \tag{3.4.3}$$

where X = observed data, Y is the predicate value and Z is the mean value of the data.

We can compare the performance of each model by computing these metrics for each one and determining which one produces the best results. Visualizing the projected values vs, the actual values can also give further insights into the models' forecasting ability.

3.6 Model push

The deployment of the trained models has not been possible at this time owing to the constraints imposed by the confidentiality of the company information and restricted access to the database management system. The statistical analysis, on the other hand, has produced useful insights and laid the groundwork for future implementation. To fully realize the model's potential, it will be necessary to solve security problems and investigate alternate ways that are consistent with the company's data regulations. The implementation of these models may be accomplished by protecting data privacy and identifying appropriate solutions, resulting in informed decision-making, and increased operational efficiency.

4. RESULTS

The results chapter provides extensive analysis and assessment of the trained models' ability to predict the intended output variable. This chapter attempts to shine a spotlight on the predictive capabilities of each model by providing insights into its performance and efficacy. Non-linear regression, Decision Tree, Random Forest, and Artificial Neural Network (ANN) are among the models being evaluated. Each model was trained using the dataset provided and will be evaluated using a variety of metrics, including Root Mean Squared Error (RMSE), Mean Square Error (MSE), and R-squared (R^2) score. To better understand the prediction accuracy of the algorithms, representations of anticipated vs actual values will be shown. The findings of this research will help to choose the best model for the situation.

4.1 Non-Linear Regression Analysis

Nonlinear regression is being utilized in this analysis to capture and model the intricate relationship between the input variables, namely Availability and Performance, and the desired output variable, Productivity. As these variables may exhibit nonlinear patterns and dependencies, nonlinear regression allows for more flexible modelling to capture complex relationships [43]. By employing nonlinear regression, the analysis aims to uncover and quantify the nonlinear dependencies between Availability and Performance and their impact on Productivity. This approach enables a more accurate prediction of Productivity and a deeper understanding of the underlying dynamics within the dataset. By leveraging the flexibility of nonlinear regression, the analysis seeks to derive insights that linear models may fail to capture, providing valuable information for decision-making and optimizing operational efficiency [44].

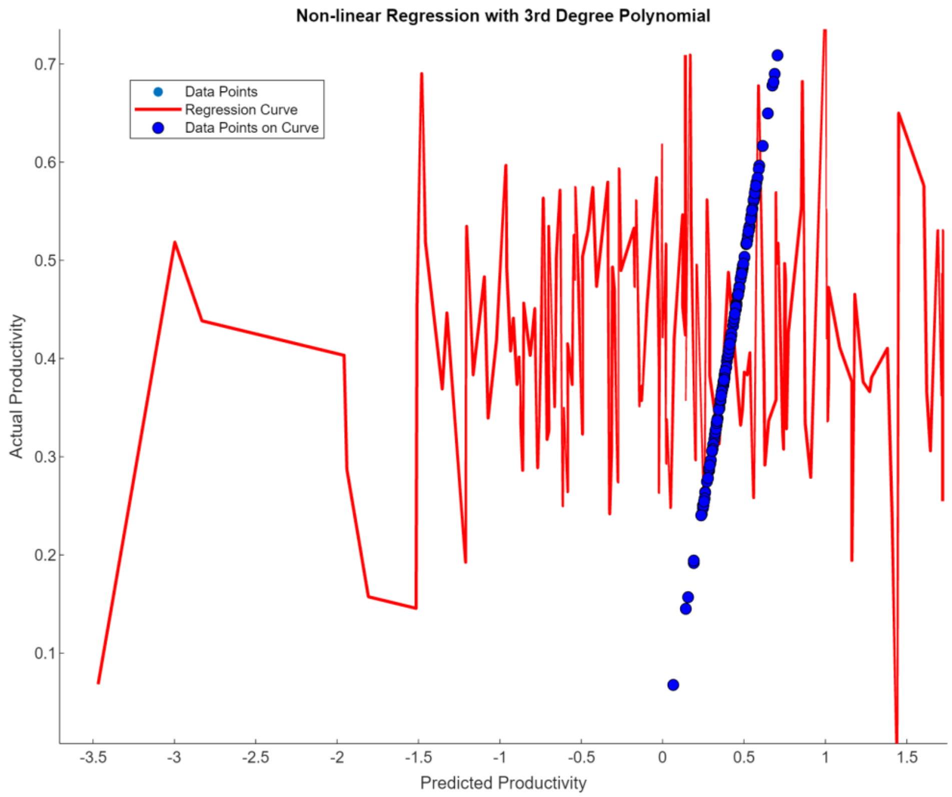

Figure 4.1.1 2D plot of non-linear regression

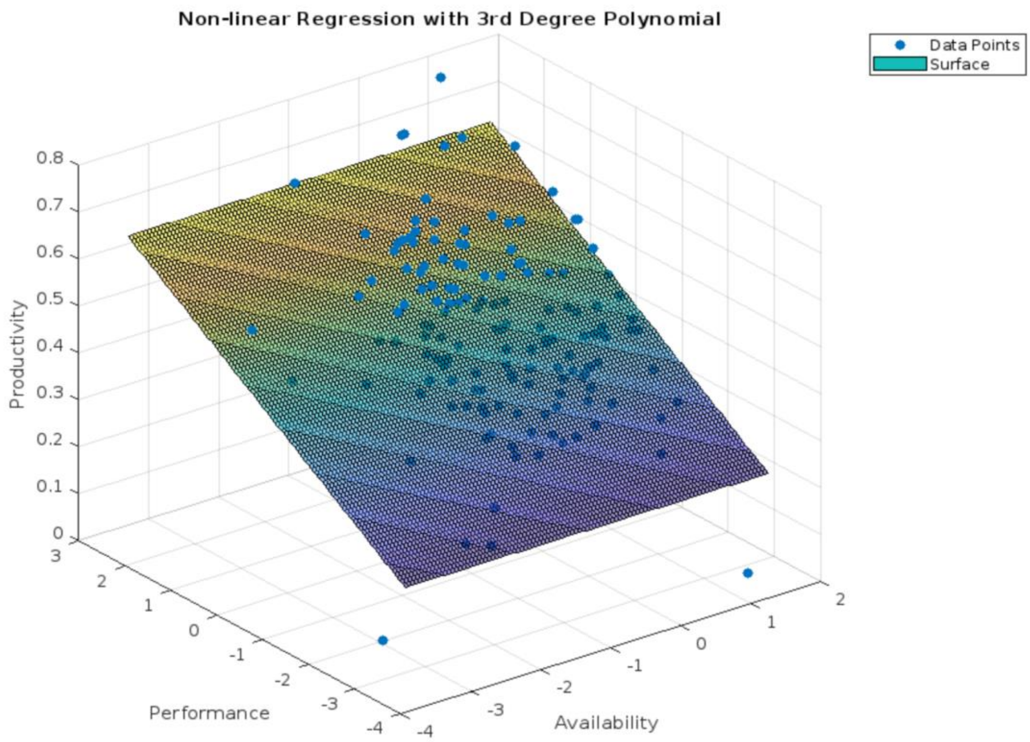

Figure 4.1.2 3D of non-linear regression

4.2 Decision tree analysis

Decision tree model using the input features (X1 and X2) which are availability and performance and the target variable productivity as (Y). It then visualizes the decision tree using the view function.

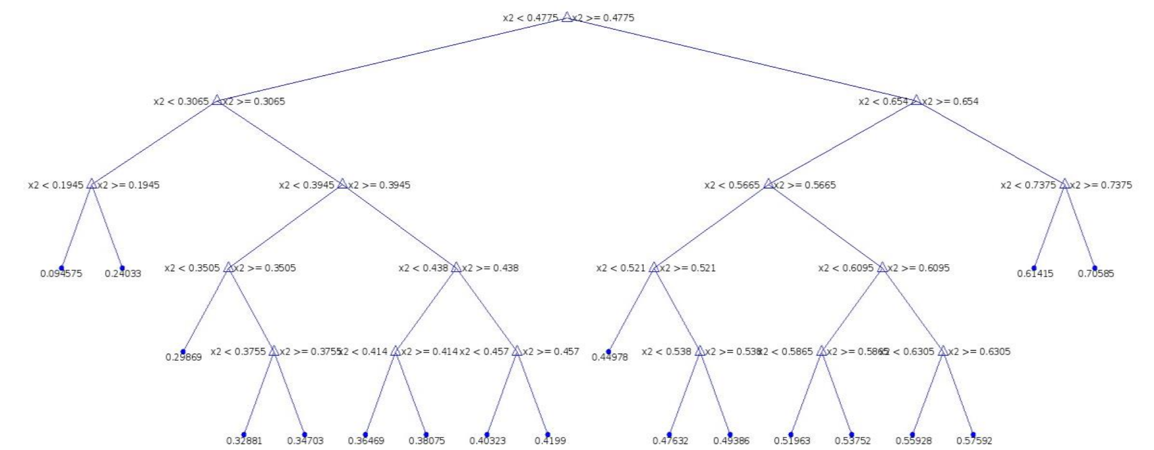

Figure 4.2.1 Output of decision tree model

The decision tree generated from the provided code relates the input features (X1 and X2) to the output variable (Y) through a series of decisions based on feature values. Starting from the root node, the tree evaluates a condition on a specific feature and compares it to a threshold value. Depending on the outcome of this comparison, the tree follows the left or right branch to traverse further down the tree. This process repeats at each internal node until a leaf node is reached.

At the leaf nodes, the decision tree provides the predicted value or class for the given input combination of X1 and X2. The path followed from the root to a particular leaf node represents the sequence of decisions made by the tree based on the input features. By examining this path, we can understand how the decision tree interprets and processes the input to arrive at the predicted output.

4.3 Random forest analysis

For the random first model, regression is used as a machine learning method. First, the data is split into Input values as X combined of both input variables and Y as the output variable. Then to train the model function called "TreeBagger" is used where the model is trained with one hundred trees. Prediction grid by creating a mesh grid of x and y values. In this case, N is set to 101, so the grid will have 101 points along each axis ranging from 0 to 1.

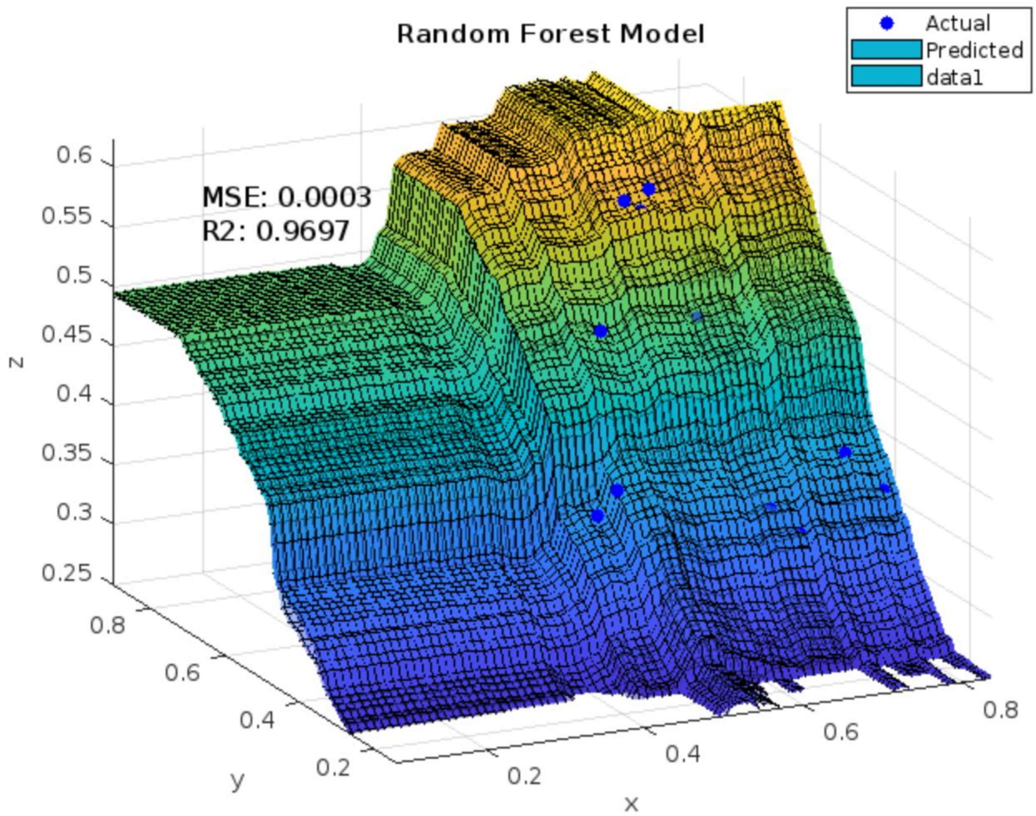

Figure 4.3.1 Surface plot of random forest model

The model predicts the response variable for each point in the prediction grid. The input to the predict function is a matrix where each row represents a point in the grid. Then a 3D surface plot of the predicted response variable (Z) against the predictor variables (x and y) is generated. The surf function is used to generate the plot, and the axis labels are set accordingly. This also shows a representation of how the predicted values of Y vary across the range of the predictor variables. The evaluation metrics (MSE and R^2) are displayed as text on the graph, providing quantitative measures of the model's performance. In this case, MES is exceptionally low which shows the predicted value is accurately related to the actual value.

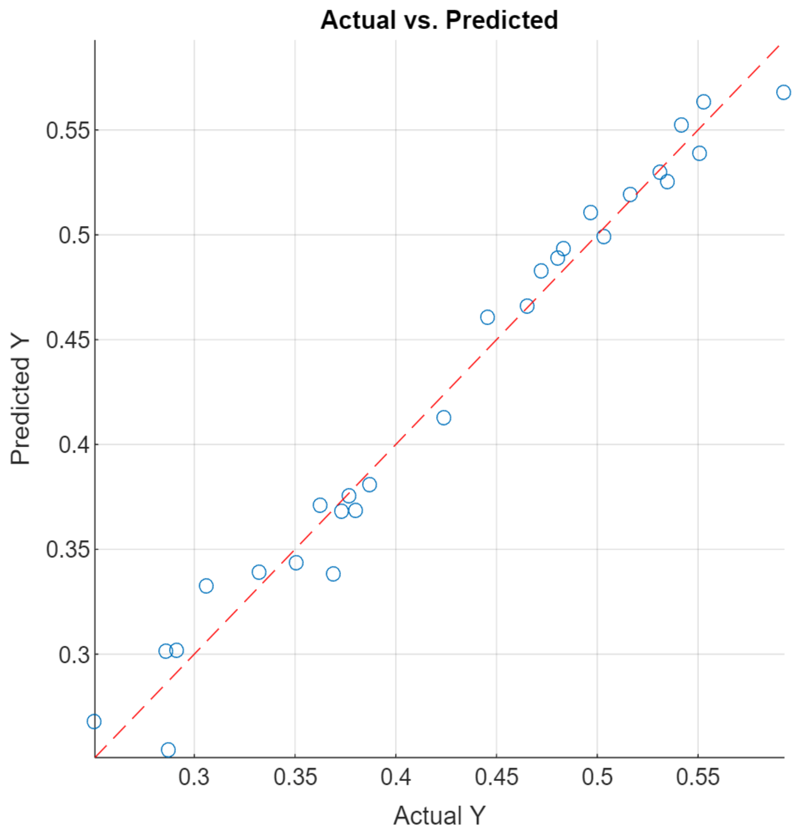

Figure 4.3.2 Scatter plot using random forest model

This graph displays a scatter plot of the actual Y values against the predicted Y values on the test set. Each point on the graph represents an instance from the test set. The x-axis represents the actual Y values, and the y-axis represents the predicted Y values. The diagonal red dashed line represents a perfect prediction, where the actual and predicted values would be identical. The points scattered around the line indicate the deviation between the actual and predicted values. As the dot density is closer to the red line, it shows that the model performed well with less error and more accuracy.

4.4 Artificial neural network analysis

An artificial neuron is essentially a biological neuron modified by engineering. It has a device with several inputs and just one output. An ANN is made up of several basic processing components that are coupled to one another and stacked as well. Like biological neurons, artificial neurons are part of artificial neural networks. These artificial neurons accept input from other components or other artificial neurons, and when the inputs are combined and weighted, the result is then translated into the output via a transfer function. The transfer function might be anything, such as a step, Sigmoid, or hyperbolic tangent function [48, 49].

An artificial neural network was applied using the input-output data from a CSV file. The input variables are separated into two columns, X1 and X2. The feedforward ANN with one hidden layer with a weight W was utilized. It is well known that for function approximation the feedforward ANN with one or two hidden layers provides high accuracy [50]. Then the output was achieved as a single neuron whose values range between o and 1.

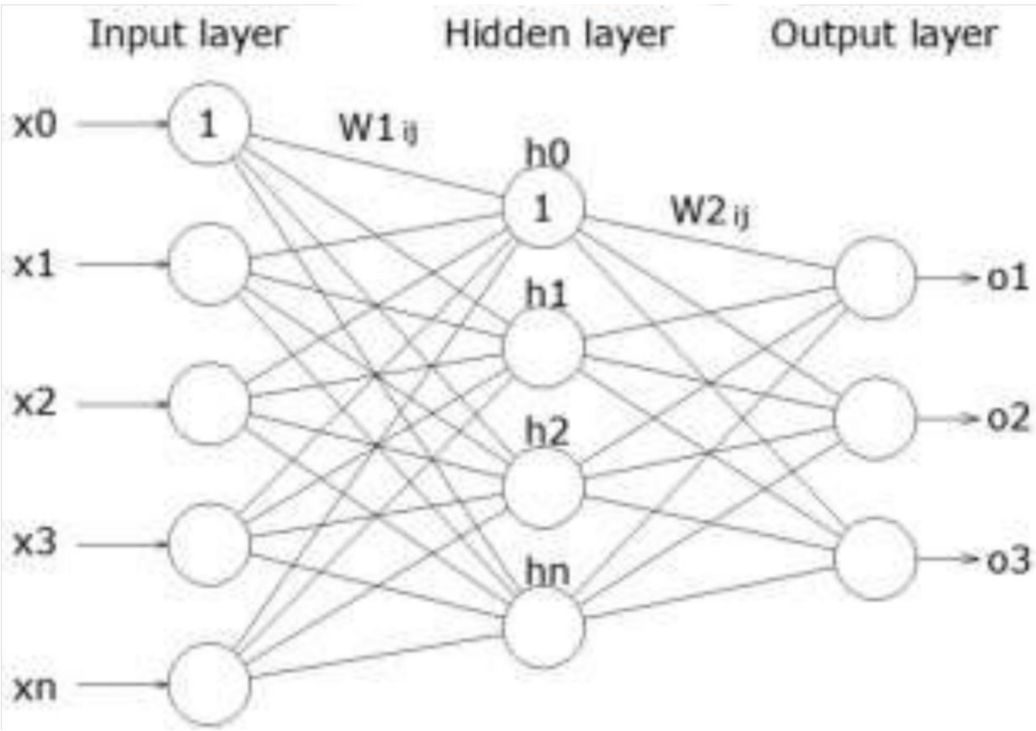

Figure 4.4.1 A feedforward ANN with multiple hidden layers [48]

After certain tuning, the hidden layer with ten neurons was found optimal (the number of neurons was increased until the mean square error decreases). The nonlinear sigmoid activation function is applied in the hidden layer and the linear function is the outer layer. The widely used value of the learning rate of 0.05 was used. The sigmoid function is achieved by [48]:

$$F(x) = \frac{1}{1+e^{-sum}} \tag{4.4.1}$$

where,

$sum = \sum_{i=0}^{n} x_i W_i$

Here x_i are the input to the neurons and W_i are the weight.

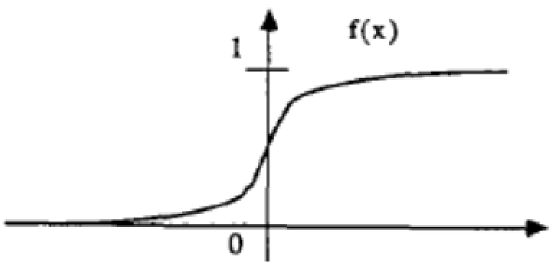

Figure 4.4.2 Sigmoid function [48]

After training, the code generates a surface plot, a contour plot, and a scatter plot to visualize the predicted output. Evaluation metrics, such as Mean Squared Error (MSE) and R-Squared (R^2), are calculated based on the predicted output and the actual output.

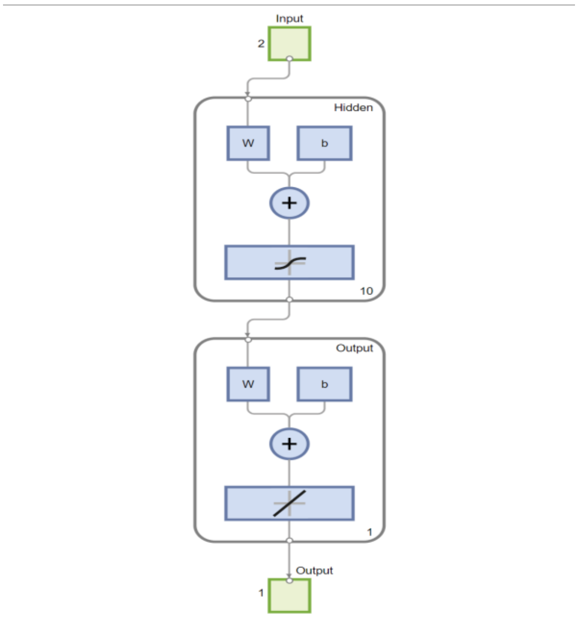

Figure 4.4.3 Network structure of ANN

Surface plot:

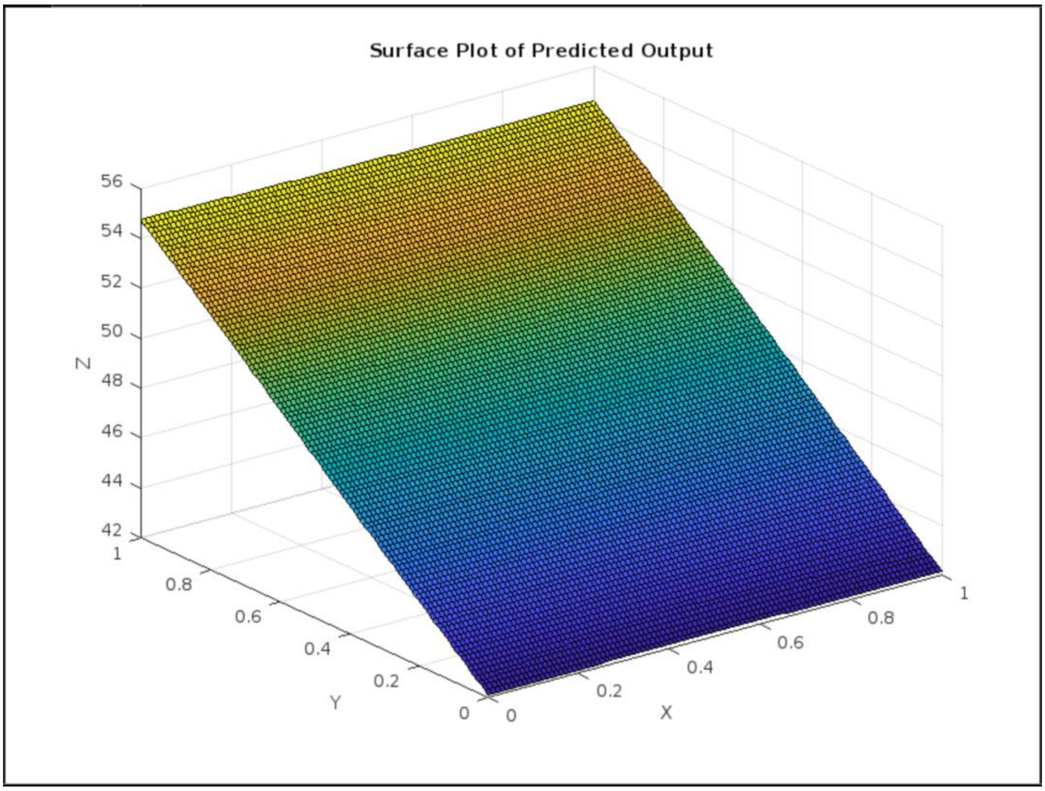

Figure 4.4.4 Surface plot of ANN

As we plotted the surface plot to visualize the pattern and trend, it shows that the plot is smooth with fewer abrupt changes in the predicted output. This smooth surface indicates a gradual and continuous change in the predicted output as the input variables change. A smooth surface suggests a more consistent relationship or dependency between the input variables and the output. Smoothness in the surface plot can be indicative of a well-trained neural network that has captured the underlying patterns and dependencies in the data. It suggests that the model has learned the relationships between the input variables and the output and can generalize its predictions effectively.

Scattered plot:

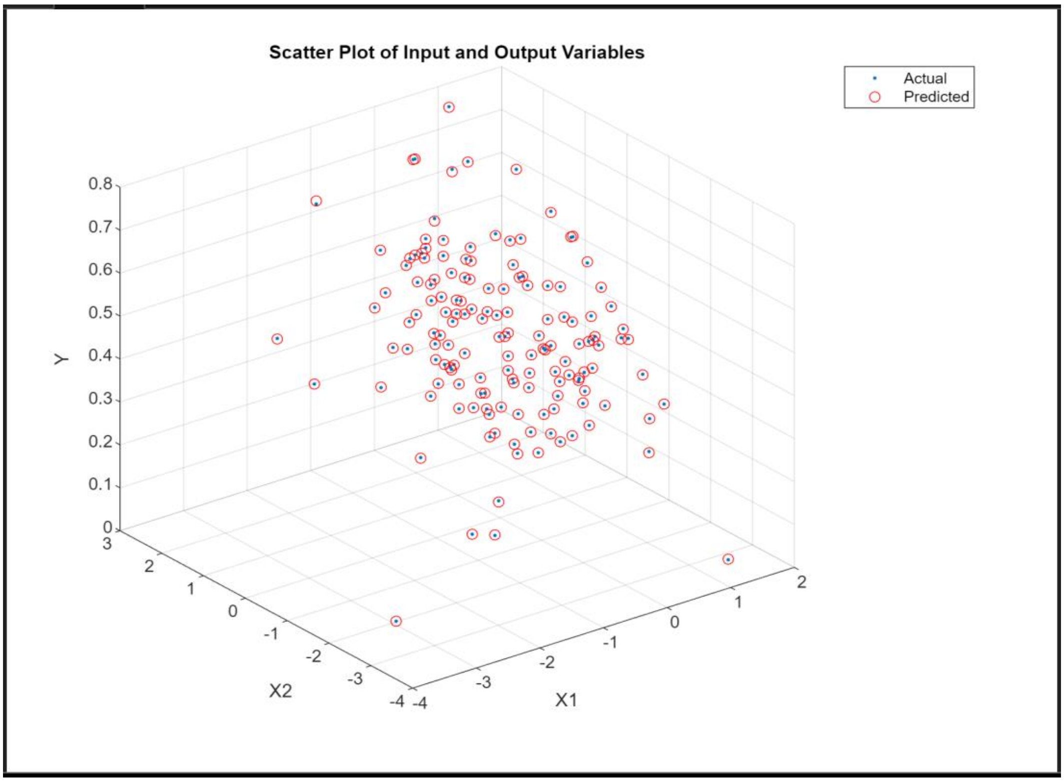

Figure 4.4.5 Scattered plot of actual productivity with predicted productivity

The scatter plot helps us assess the model's performance by visually comparing the predicted output to the actual output. It can be seen how well the model is capturing the relationships between the input and output variables and help identify any discrepancies or patterns in the predictions.

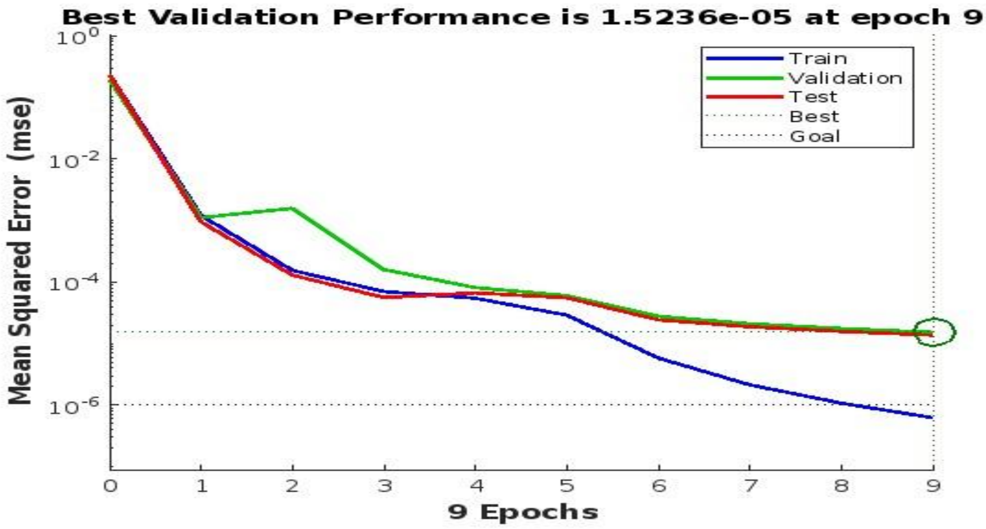

Figure 4.4.6 Performance graph of ANN

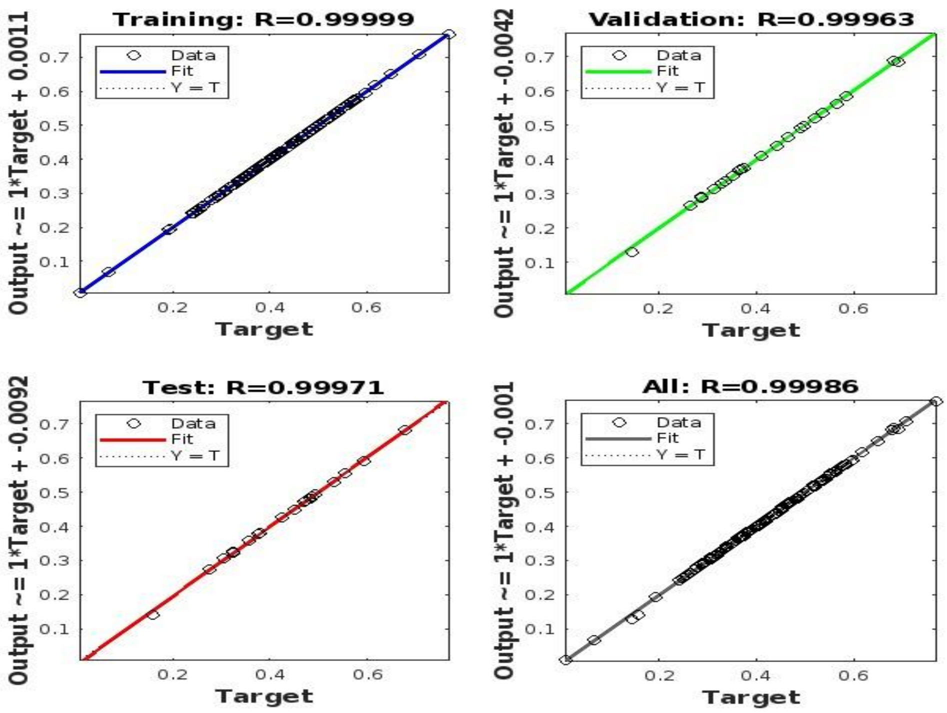

Figure 4.4.7 Regression graph of ANN

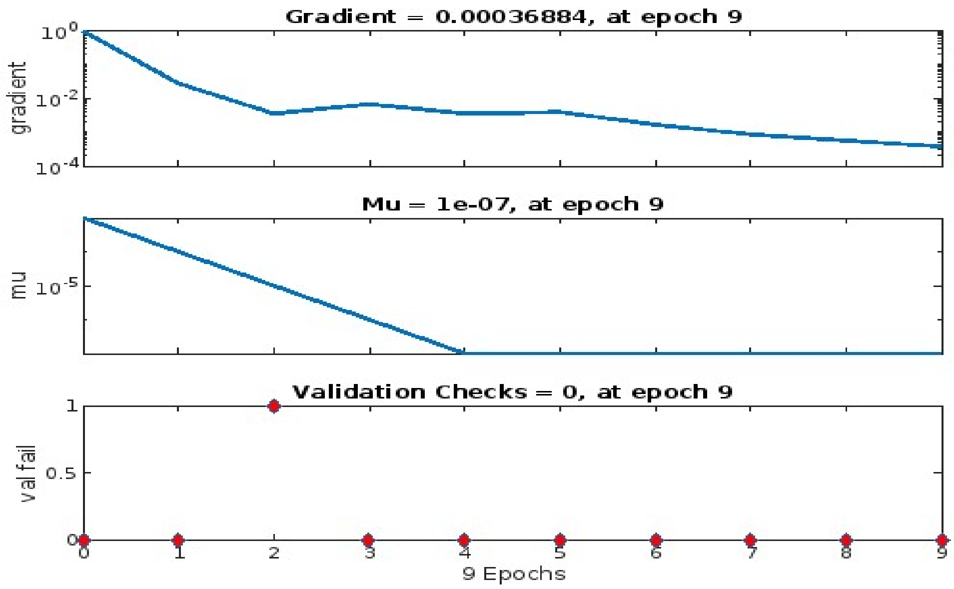

Figure 4.4.8 Training state graph of ANN

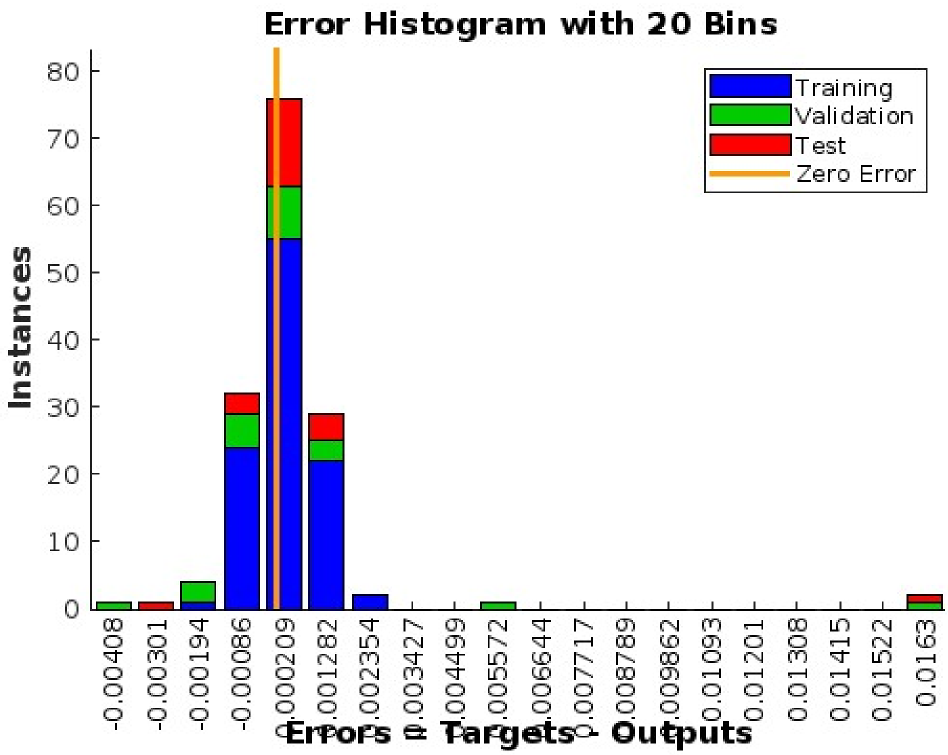

Figure 4.4.9 Training error graph of ANN

For each epoch, we modify the program the calculate the training time, loss, MSE and R square to analyse the performance.

Table 4.4-1 Performance indicator of ANN

Epoch	Training Time	Loss	MSE	R-square
1	3,1038	0,053866	3,3008e-06	0,99979
2	3,1893	0,0021401	3,3008e-06	0,99979
3	3,2042	0,00069931	3,3008e-06	0,99979
4	3,2171	0,00028029	3,3008e-06	0,99979
5	3,2294	9,6018e-05	3,3008e-06	0,99979
6	3,2577	3,1112e-06	3,3008e-06	0,99979
7	3,2599	2,1419e-06	3,3008e-06	0,99979
8	3,2613	1,8209e-06	3,3008e-06	0,99979
9	3,2631	1,5756e-06	3,3008e-06	0,99979

In the context of the provided input data, which consists of availability and performance, and the output variable of productivity, the R-squared (R^2) score serves as a measure of how well the regression model fits the observed data. The R-squared score quantifies the proportion of the variability in the productivity variable that is explained by the availability and performance variables. The training R-squared indicates the model's fit to the training data, while the validation and test R-squared scores assess its generalization performance on unseen data. The overall R-squared score combines the performance across all datasets, offering an overall evaluation of the model's predictive capability. It is important to consider both high R-squared values and the risk of overfitting when interpreting the results.

4.5 Comparing performance

A comparison matrix is created to evaluate and compare the performance of different models for the given dataset. Four models are considered: Polynomial Regression, Decision Tree, Random Forest, and Artificial Neural Network (ANN). The evaluation metrics used were the mean squared error (MSE), root mean square error (RMSE) and the R-squared (R^2) coefficient.

Table 4.5-1 Performance comparison of the trained model

Model	**MSE**	**RMSE**	R^2
Polynomial regression	0,013378	0,11567	-0,13352
Decision Tree	0,00011776	0,010852	0,99002
Random Forest	0,0036015	0,060013	0,69485
ANN	6,9407e-08	0,00026345	0,99999

The results show that the Polynomial Regression model has an MSE of 0,013378 and an RMSE of 0,11567. However, it performs poorly with a negative R^2 value of -0,13352.

On the other hand, the Decision Tree model exhibits excellent performance with an extremely low MSE of 0,00011776 and a correspondingly low RMSE of 0,010852. It achieves a high R^2 value of 0,99002, indicating that it explains 99% of the variance in the target variable.

The Random Forest model performs well with an MSE of 0,0036015 and an RMSE of 0,060013. It achieves a R^2 value of 0,69485, indicating moderate predictive power.

The Artificial Neural Network (ANN) model outperforms all other models with an exceptionally low MSE of 6,9407e-08 and an impressively low RMSE of 0,00026345. It achieves a near-perfect square value of 0,99999, demonstrating a strong linear relationship and excellent predictive accuracy.

Despite the limited data points provide four traditional machine-learning models are trained. The performance of all the models was satisfactory for this paper. With the larger data set the outcome would have been more accurate and prediction would be better and more accurate. From all the graphs and plots, it shows that the predicted value corresponds well to the actual value.

To conclude, the Decision Tree and ANN models show superior performance compared to the Polynomial Regression and Random Forest models. The Decision Tree model stands out with its extremely low MSE and RMSE values, while the ANN model highlights exceptional accuracy and a near-perfect R^2 value. For the analysis of such manufacturing-related data, ANN is the best machine-learning method, to begin with. Comparing various methods provide more insight into how the input data is related to the actual and predicted output. Based on the analysis, SEMs can identify bottlenecks that result in lower availability and reduced performance. By eliminating such bottlenecks in production, productivity is increased. Additionally, various outputs can be simulated by altering the input data by estimating and predict on how such improvements impact total productivity.

SUMMARY

The goals of this thesis were achieved by building a solid foundation for SMEs transitioning to smart manufacturing with the use of traditional machine learning algorithms on the production matrices data collected from a machine in SMEs. This thesis paper demonstrates the need for data-driven insights and prediction models in the manufacturing industry, particularly for small and medium-sized enterprises (SMEs) with limited resources. By leveraging data analysis, big data analytics, and machine learning, this thesis aims to predict the outcome in SMEs. The theoretical framework explores production metrics, popular machine learning models, and their applications, emphasizing their potential to facilitate data-driven decision-making and uncover hidden correlations within the manufacturing sector.

The research methodology described in detail in the third chapter covered data collection, data cleaning processes, data transformation methods, model selection, and model analysis and assessment. The emphasis is placed on collecting relevant and trustworthy data, ensuring data quality and integrity, and preparing the data for subsequent model training and selection. The fourth chapter presents the findings of the research, with a focus on production metrics in SMEs. Non-linear regression analysis is employed to reveal connections between variables, while decision tree analysis identifies common decision patterns and significant variables influencing outcomes in SME production metrics. Random forest analysis evaluates the combined impact of multiple factors on the objective. In the specific case of SMEs, artificial neural network analysis demonstrates its ability to recognize complex patterns and provide accurate forecasts. Through comprehensive model comparisons, considering accuracy, effectiveness, and adaptability to dynamic challenges, valuable insights are gained regarding the application of machine learning techniques in the industrial sector, specifically for production metrics in SMEs.

Various traditional machine learning models were trained and evaluated using limited data points. Despite the limitations, the performance of all the models was satisfactory for the research. However, it is important to note that with a larger and more diverse dataset, the accuracy and prediction capabilities of these models would likely improve significantly. The analysis of graphs and plots revealed a strong correspondence between the predicted values and the actual values.

Among the models tested, the Decision Tree and Artificial Neural Network (ANN) models demonstrated superior performance compared to Polynomial Regression and Random Forest models. The Decision Tree model stood out with its remarkably low Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values, indicating its high precision. On the other hand, the ANN model showcased exceptional accuracy and a near-perfect R^2 value, further highlighting its effectiveness.

For the analysis of manufacturing-related data, the ANN model emerged as the best machine-learning method to initiate the process. By comparing various methods, valuable insights were gained regarding the relationship between input data and the actual and predicted output. These insights enabled the identification of bottlenecks that impede availability and reduce performance in production processes. By eliminating such bottlenecks, productivity can be increased. Furthermore, by simulating various output scenarios through estimation and prediction, the impact of improvements on overall productivity can be assessed.

However, future work should address the limitations by expanding the dataset to include a larger and more diverse set of production metrics from various SMEs, enabling more accurate predictions and a better understanding of correlations. Further research should explore alternative machine learning models and ensemble techniques, such as support vector machines, gradient boosting, and deep learning architectures, to improve performance and prediction accuracy. Additionally, incorporating advanced analytical techniques, including anomaly detection and outlier analysis, would help identify exceptional cases in production processes and enhance intervention strategies.

Integration of domain-specific knowledge and expert insights into machine learning models is essential for improved interpretability and performance. Collaborations with industrial experts can provide valuable input to tailor the models to the specific needs and challenges of the manufacturing industry. To ensure practical implementation, future work should focus on the deployment of the developed models in real-world manufacturing settings. Considerations such as model scalability, computational efficiency, and user-friendly interfaces should be addressed. Validating the models' performance and impact through pilot studies or industrial trials will provide evidence of their effectiveness and applicability in SMEs.

KOKKUVÕTE

Käesoleva lõputöö eesmärgid saavutati, luues kindla aluse VKE-de üleminekuks nutikale tootmisele, kasutades traditsioonilisi masinõppe algoritme ja VKE-des kogutud tootmise andmeid. Vaadeldav lõputöö toob välja andmetel põhinevate teadmiste ja ennustusmudelite kogumise vajaduse tootmissektoris, eriti piiratud ressurssidega väike- ja keskmise suurusega ettevõtetes (VKE-d). Andmeanalüüsi, suurandmeanalüütika ja masinõppe abil püüab antud lõputöö ennustada tootmistulemusi VKE-des. Töö teoreetiline raamistik uurib tootmisnäitajaid, populaarseid masinõppe mudeleid ja nende rakendusi, rõhutades nende potentsiaali andmetel põhineva otsustamise toetamiseks ning varjatud seoste avastamiseks tootmissektoris.

Töö kolmandas peatükis kirjeldatud uurimismetoodika hõlmas üksikasjalikult andmete kogumist, andmete puhastamise protsesse, andmete transformeerimismeetodeid, mudeli valikut ning mudeli analüüsi ja hindamist. Rõhk on pandud asjakohaste ja usaldusväärsete andmete kogumisele, andmete kvaliteedi ja terviklikkuse tagamisele ning nende ettevalmistamisele järgnevaks mudelite õpetamiseks ja valikuks. Töö neljas peatükk esitleb uurimistöö tulemusi, keskendudes VKE-de tootmisnäitajatele. Seoste leidmiseks muutujate vahel kasutatakse mittelineaarset regressioonianalüüsi, samal ajal kui otsustuspuuanalüüs tuvastab tavalised otsustusmustrid ja olulised muutujad, mis mõjutavad VKE tootmisnäitajate tulemusi. Juhumetsade analüüs hindab mitme teguri kombineeritud mõju eesmärgile. VKE-de konkreetse juhtumi puhul näitab tehisnärvivõrgu analüüs selle võimet ära tunda keerukaid mustreid ja anda täpsed prognoosid. Mudelite võrdluste abil, arvestades täpsust, efektiivsust ja dünaamiliste väljakutsetega seotud kohanemisvõimet saadakse väärtuslikke teadmisi masinõppe tehnikate rakendamise võimaluste kohta tööstussektoris, eriti VKE-de tootmisnäitajate osas.

Erinevaid traditsioonilisi masinõppe mudeleid treeniti ja hinnati piiratud andmepunktide abil. Vaatamata piirangutele oli kõikide mudelite jõudlus uurimistöö jaoks rahuldav. Siiski on oluline märkida, et suurema ja mitmekesisema andmekogumi korral suureneks nende mudelite täpsus ja ennustusvõime oluliselt. Graafikute ja jooniste analüüs näitas tugevat vastavust ennustatud väärtuste ja tegelike väärtuste vahel.

Testitud mudelite hulgas näitasid otsustuspuu ja kunstliku närvivõrgu (ANN) mudelid ülejäänutega võrreldes paremat jõudlust. Otsustuspuu mudel eristus eriti madala keskmise ruutveaga (MSE) ja juuritud keskmise ruutveaga (RMSE) väärtustega. Teisalt näitas tehisnärvivõrgu mudel erakordset täpsust ja peaaegu ideaalset R^2 väärtust, rõhutades selle tõhusust.

Tootmisega seotud andmete analüüsis osutus kunstliku närvivõrgu mudel parimaks masinõppe meetodiks protsessi käivitamiseks. Erinevate meetodite võrdlemise kaudu saadi väärtuslikke teadmisi sisendandmete ning tegeliku ja ennustatud väljundi vahelisest seosest. Need teadmised võimaldasid tuvastada kitsaskohti, mis takistavad kättesaadavust ja vähendavad tootmisprotsesside tõhusust. Kitsaskohtade kõrvaldamisega saab suurendada tootlikkust. Lisaks saab erinevate väljundstsenaariumide simuleerimise kaudu hinnata parenduste mõju üldisele tootlikkusele läbi hinnangute ja prognooside.

Tulevased uurimistööd peaksid aga käsitlema piiranguid, laiendades andmekogumit, et hõlmata suurem ja mitmekesisem valik tootmisnäitajaid erinevatest VKE-dest, võimaldades täpsemaid prognoose ja paremat korrelatsioonide mõistmist. Edasine uurimistöö peaks uurima alternatiivseid masinõppe mudeleid ja ansamblitehnikaid, nagu toetusvektormasinad, gradienttõstmine ja süvauuringu arhitektuurid, et parandada jõudlust ja ennustuste täpsust. Lisaks aitaksid täiustatud analüütilised tehnikad, sealhulgas ebanormaalsuste tuvastamine ja väljajääja analüüs tuvastada erandlikke juhtumeid tootmisprotsessides ning parandada sekkumisstrateegiaid.

Valdkonnapõhiste teadmiste ja ekspertarvamuste integreerimine masinõppe mudelitesse on oluline parema tõlgendatavuse ja jõudluse saavutamiseks. Koostöö tööstusekspertidega võib pakkuda väärtuslikke sisendeid mudelite kohandamiseks vastavalt tootmisvaldkonna konkreetsetele vajadustele ja väljakutsetele. Praktilise rakenduse tagamiseks peaks tulevane töö keskenduma välja arendatud mudelite kasutuselevõtule reaalsetes tootmisoludes. Tuleks arvestada mudeli skaleeritavust, arvutuslikku tõhusust ja kasutajasõbralikke liideseid. Mudelite jõudluse ja mõju valideerimine pilootuuringute või tööstuskatsete kaudu annab tõendeid nende efektiivsuse ja rakendatavuse kohta VKE-des.

REFERENCES

[1] Mittal, S., Khan, M. A., Romero, D., & Wuest, T. (2017). Smart manufacturing: Characteristics, technologies and enabling factors. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *233*(5), 1342–1361. https://doi.org/10.1177/0954405417736547

[2] Bianchini, M., & Michalkova, V. (2019). Data Analytics in SMEs. *OECD SME and Entrepreneurship Papers*.

[3] Maroufkhani, P., Iranmanesh, M., & Ghobakhloo, M. (2022). Determinants of big data analytics adoption in small and medium-sized enterprises (SMEs). *Industrial Management & Data Systems*, *123*(1), 278–301. https://doi.org/10.1108/imds-11-2021-0695

[4] Han, H., & Trimi, S. (2022). Towards a data science platform for improving SME collaboration through Industry 4.0 Technologies. *Technological Forecasting and Social Change*, *174*, 121242.

[5] Hansen, E. B., & Bøgh, S. (2021). Artificial Intelligence and Internet of things in small and medium-sized enterprises: A survey. *Journal of Manufacturing Systems*, *58*, 362–372. https://doi.org/10.1016/j.jmsy.2020.08.009

[6] Czvetkó, T., Kummer, A., Ruppert, T., & Abonyi, J. (2022). Data-driven business process management-based development of Industry 4.0 Solutions. *CIRP Journal of Manufacturing Science and Technology, 36,* 117–132. https://doi.org/10.1016/j.cirpj.2021.12.002

[7] Nacchia, M., Fruggiero, F., Lambiase, A., & Bruton, K. (2021). A systematic mapping of the advancing use of machine learning techniques for predictive maintenance in the manufacturing sector. *Applied Sciences, 11(6), 2546.* https://doi.org/10.3390/app11062546

[8] Han, H., & Trimi, S. (2022). Towards a data science platform for improving SME collaboration through Industry 4.0 technologies. *Technological Forecasting and Social Change*, *174*, 121242.

[9] Georgios, Lampropoulos, Siakas Kerstin, and Anastasiadis Theofylaktos. "Internet of things in the context of industry 4.0: An overview." (2019).

[10] Kang, H. S., Lee, J. Y., Choi, S., Kim, H., Park, J. H., Son, J. Y., ... & Noh, S. D. (2016). Smart manufacturing: Past research, present findings, and future directions. *International journal of precision engineering and manufacturing-green technology*, *3*, 111-128.

[11] Kusiak, A. (2017). Smart manufacturing must embrace big data. *Nature*, *544*(7648), 23-25.

[12] Phuyal, S., Bista, D., & Bista, R. (2020). Challenges, opportunities and future directions of smart manufacturing: a state of art review. *Sustainable Futures*, *2*, 100023.

[13] Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*, *48*, 157-169.

[14] Haapala, K. R., Zhao, F., & Camelio, J. A. (2018). Design for sustainability: a review and future directions for the built environment. *Journal of Cleaner Production, 178,* 229-239.

[15] Liu, L., & Zhang, Y. (2019). Digital manufacturing for sustainable development: A review. *Journal of Cleaner Production, 230*, 301-313.

[16] Sarkis, J., Zhu, Q., & Lai, K. H. (2011). An organisational theoretic review of green supply chain management literature. *International journal of production economics, 130(1)*, 1-15.

[17] Dabbagh, M., Pourghannad, B., & Bemanian, M. R. (2021). Resilient manufacturing systems in Industry 4.0: A systematic literature review. *Journal of Manufacturing Systems, 58*, 1-23.

[18] Kusiak, A. (2019). Smart manufacturing: Past research, present findings, and future directions. *Journal of Manufacturing Systems, 51*, 111-131.

[19] Li, X., Wang, L., Xu, X., & Li, Y. (2020). A review on the applications of digital twin in the context of Industry 4.0. *Journal of Manufacturing Systems, 56*, 221-237.

[20] D. Dey, A. Mukherjee, and R. Kumar, Industry 4.0 and its impact on SMEs: A review, *Procedia Manufacturing, vol. 20*, pp. 131-138, 2018.

[21] R. P. Singh, P. Singh, and R. K. Sahu, Industry 4.0 and Its Applications in Smart Manufacturing: A Review, *International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET), 2020*, pp. 97-101.

[22] European Commission, "2019 Report on the Implementation of the Commission Communication on "Investing in a smart, innovative and sustainable Industry: A renewed EU Industrial Policy Strategy", *European Commission*, 2019.

[23] T. AlGeddawy, E. ElMeligy, and M. S. AlSalhi, Challenges and solutions of implementing industry 4.0 in small and medium-sized enterprises, *2nd International Conference on Advances in Mechanical Engineering and its Interdisciplinary Areas (ICAMEI*), 2018, pp. 1-6.

[24] European Commission, "European Digital Innovation Hubs," European Commission,[Online].Available:https://ec.europa.eu/commission/presscorner/detail/en/fs_20_426 [Accessed: Apr. 10, 2023].

[25] A. Gaaloul, M. Kebairi, and J. Mendling, Industry 4.0 adoption in small and medium-sized enterprises: An organizational culture perspective, *Journal of Enterprise Information Management, vol. 31, no. 4*, pp. 642-660, 2018.

[26] Kang, Z., Catal, C., & Tekinerdogan, B. (2020). Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering*, *149*, 106773.

[27] Wang, B., Tao, F., Fang, X., Liu, C., Liu, Y., & Freiheit, T. (2021). Smart manufacturing and intelligent manufacturing: A comparative review. *Engineering*, *7*(6), 738-757.

[28] Dai, H. N., Wang, H., Xu, G., Wan, J., & Imran, M. (2020). Big data analytics for manufacturing internet of things: opportunities, challenges and enabling technologies. *Enterprise Information Systems*, *14*(9-10), 1279-1303.

[29] Wang, L., Törngren, M., & Onori, M. (2015). Current status and advancement of cyber-physical systems in manufacturing. *Journal of Manufacturing Systems*, *37*, 517-527.

[30] Mourtzis, D., Vlachou, E., & Milas, N. (2018). A review on predictive maintenance of production systems using sensors. *IEEE Access*, 6, 20606-20625.

[31] Wu, D., & Rosen, D. W. (2019). *Artificial intelligence in manufacturing*. Springer.

[32] Zadeh, S. H., & Pan, Y. (2021). Artificial intelligence applications in manufacturing: A literature review. *Computers & Industrial Engineering*, 152, 107013.

[33] Wang, B., Tao, F., Fang, X., Liu, C., Liu, Y., & Freiheit, T. (2021). Smart manufacturing and intelligent manufacturing: A comparative review. *Engineering*, 7(6), 738-757.

[34] Shang, C., & You, F. (2019). Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era. *Engineering*, *5*(6), 1010-1016.

[35] Rai, R., Tiwari, M. K., Ivanov, D., & Dolgui, A. (2021). Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research*, *59*(16), 4773-4778.

[36] Kalita, J. (2022). *Machine Learning*. Milton: CRC Press LLC.

[37] Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR). [Internet]*, *9*, 381-386.

[38] Huynh, V. N., Inuiguchi, M., Le, B., Le, B. N., & Denoeux, T. (Eds.). (2016). *Integrated Uncertainty in Knowledge Modelling and Decision Making: 5th International Symposium, IUKM 2016, Da Nang, Vietnam, November 30-December 2, 2016, Proceedings* (Vol. 9978). Springer.

[39] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*(3), 210-229.

[40] El Naqa, I., & Murphy, M. J. (2015). *What is machine learning?* (pp. 3-11). Springer International Publishing.

[41] ung, A. (2022). *Machine Learning* (Machine Learning. Foundations, Methodologies, and Applications). Singapore: Springer.

[42] Chen, T., Sampath, V., May, M., Shan, S., Jorg, O., Aguilar Martín, J., . . . Calaon, M. (2023). Machine Learning in Manufacturing towards Industry 4.0: From 'For Now' to 'Four-Know'. *Applied Sciences, 13*(3), 1903.

[43] Bertolini, M., Mezzogori, D., Neroni, M., & Zammori, F. (2021). Machine Learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications, 175*, 114820.

[44] Ouadah, A., Zemmouchi-Ghomari, L., & Salhi, N. (2022). Selecting an appropriate supervised machine learning algorithm for predictive maintenance. *International Journal of Advanced Manufacturing Technology, 119*(7-8), 4277-4301.

[45] Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524.

[46] The MathWorks, Inc. (2023). MATLAB version: 9.14.0 (R2023a). Available: https://www.mathworks.com.

[47] Oytun, M., Tinazci, C., Sekeroglu, B., Acikada, C., & Yavuz, H. (2020). Performance Prediction and Evaluation in Female Handball Players Using Machine Learning Models. *IEEE Access, 8*, 116321-116335.

[48] Kukreja, H., Bharath, N., Siddesh, C. S., & Kuldeep, S. (2016). An introduction to artificial neural network. *Int J Adv Res Innov Ideas Educ*, *1*, 27-30.

[49] Sharma, V., Rai, S., & Dev, A. (2012). A comprehensive study of artificial neural networks. *International Journal of Advanced research in computer science and software engineering*, *2*(10).

[50] Sheela, K., & Deepa, S. (2013). Review on Methods to Fix Number of Hidden Neurons in Neural Networks. *Mathematical Problems in Engineering, 2013*, 1-11.

[51] OpenAI. (2021). ChatGPT (version GPT-3.5) [Software]. Retrieved from https://openai.com/

APPENDIX

A3.1 Data structure:

Date	Workstation	Availability	Perf ormen	Quality	OEE	TEEP	Result	Productivity	Op active	Off time	Short stop	Long Stop	Working time	Total time
2022/12/01	Production unit 1	36%	45%	100%	16%	8.2%	10630 pcs	40.35 pcs/min	00:00:00	10:04:16	02:04:11	07:28:05	04:23:27	24:00:00
2022/12/02	Production unit 1	56%	50%	100%	28%	17%	22489 pcs	45.07 pcs/min	00:00:00	07:09:30	01:25:42	07:05:50	08:18:57	24:00:00
2022/12/03	Production unit 1	0%	0%	100%	0%	0%	0 pcs	0 pcs/min	00:00:00	24:00:00	00:00:00	00:00:00	00:00:00	24:00:00
2022/12/04	Production unit 1	0%	0%	100%	0%	0%	0 pcs	0 pcs/min	00:00:00	24:00:00	00:00:00	00:00:00	00:00:00	24:00:00
2022/12/05	Production unit 1	55%	37%	100%	20%	13%	16353 pcs	33.41 pcs/min	00:00:00	07:01:21	01:56:43	06:52:28	08:09:27	24:00:00
2022/12/06	Production unit 1	68%	39%	100%	26%	9%	11212 pcs	35.14 pcs/min	00:00:00	15:00:25	00:42:08	02:58:24	05:19:03	24:00:00
2022/12/07	Production unit 1	36.3%	32%	100%	12%	5.9%	7641 pcs	28.74 pcs/min	00:00:00	10:03:04	01:01:34	08:29:30	04:25:51	24:00:00
2022/12/08	Production unit 1	72.5%	47%	100%	34%	21.1%	27346 pcs	42.42 pcs/min	00:00:00	07:03:48	01:14:37	04:56:55	10:44:39	24:00:00
2022/12/09	Production unit 1	55.9%	45%	100%	25%	15%	19970 pcs	40.33 pcs/min	00:00:00	07:07:43	01:10:22	07:26:46	08:15:09	24:00:00
2022/12/10	Production unit 1	0%	0%	100%	0%	0%	0 pcs	0 pcs/min	00:00:00	24:00:00	00:00:00	00:00:00	00:00:00	24:00:00
2022/12/11	Production unit 1	0%	0%	100%	0%	0%	0 pcs	0 pcs/min	00:00:00	23:59:55	00:00:05	00:00:00	00:00:00	24:00:00
2022/12/12	Production unit 1	58%	35%	100%	20%	13%	16224 pcs	31.81 pcs/min	00:00:00	07:09:06	02:16:31	06:04:20	08:30:02	24:00:00
2022/12/13	Production unit 1	69%	64.9%	100%	45%	28%	35830 pcs	58.41 pcs/min	00:00:00	07:09:22	01:05:44	05:31:27	10:13:27	24:00:00
2022/12/14	Production unit 1	73%	79%	100%	58%	35%	44739 pcs	70.88 pcs/min	00:00:00	07:32:18	00:26:41	05:29:47	10:31:13	24:00:00
2022/12/15	Production unit 1	82%	63.2%	100%	52%	32%	41516 pcs	56.88 pcs/min	00:00:00	07:05:38	00:58:28	03:45:59	12:09:54	24:00:00
2022/12/16	Production unit 1	65%	55%	100%	35%	22%	28060 pcs	49.28 pcs/min	00:00:00	07:13:55	02:07:00	05:09:40	09:29:25	24:00:00
2022/12/17	Production unit 1	0%	0%	100%	0%	0%	0 pcs	0 pcs/min	00:00:00	24:00:00	00:00:00	00:00:00	00:00:00	24:00:00

A4.1 Code for polynomial regression:

```
% Load the data from the CSV file
data = readtable('Data.csv');
X = table2array(data(:, 13:14));
Y = table2array(data(:, 16));

% Calculate column means
X_mean = mean(X, 'omitnan');
Y_mean = mean(Y, 'omitnan');

% Replace missing values with column means
X(isnan(X)) = X_mean;
Y(isnan(Y)) = Y_mean;

% Normalize the input variables
X_normalized = normalize(X);
X1 = X_normalized(:, 1); % Column 13 - Availability
X2 = X_normalized(:, 2); % Column 14 - Performance

% Create polynomial features up to the 3rd degree
X_poly = [X1, X2, X1.^2, X2.^2, X1.^3, X2.^3];

% Fit a linear regression model
model = fitlm(X_poly, Y);

% Obtain the predicted values
Y_pred = predict(model, X_poly);

% Calculate evaluation metrics
mse = mean((Y - Y_pred).^2, 'omitnan');
r2 = 1 - sum((Y - Y_pred).^2) / sum((Y - mean(Y)).^2);

% Create a table of evaluation metrics
evaluationTable = table(mse, r2, 'VariableNames', {'MSE', 'R2'});

% Display the evaluation table
disp(evaluationTable);

% Create a scatter plot of the input and output variables
figure;
scatter(Y_pred, Y, 'filled');
hold on;

% Sort the data for plotting the regression curve
[X_sorted, sortIdx] = sort(X1);
Y_sorted = Y_pred(sortIdx);

% Plot the regression curve
plot(X_sorted, Y_sorted, 'r-', 'LineWidth', 2);

% Plot the data points along the regression curve
plot(Y_pred, Y, 'ko', 'MarkerSize', 7, 'MarkerFaceColor', 'b');

xlabel('Predicted Productivity');
ylabel('Actual Productivity');
title('Non-linear Regression with 3rd Degree Polynomial');
legend('Data Points', 'Regression Curve', 'Data Points on Curve');
```

A4.2 Code for random forest model:

```
% Load the data from the CSV file
data = readmatrix('Data.csv');

% Split the data into predictor variables (X) and response variable (Y)
X = data(:, 13:14);
Y = data(:, 16);

% Split the predictor variables X into two separate arrays: X1 and X2
X1 = X(:, 1); % Column 13 - Availability
X2 = X(:, 2); % Column 14 - Performance

% Split X1 and X2 into train and test sets (80% train, 20% test)
cv = cvpartition(size(X1, 1), 'HoldOut', 0.2); % 80% train, 20% test
trainInd = training(cv);
testInd = test(cv);
X1_train = X1(trainInd);
X1_test = X1(testInd);
X2_train = X2(trainInd);
X2_test = X2(testInd);

% Split the response variable Y into train and test sets
Y_train = Y(trainInd);
Y_test = Y(testInd);

% Train the model
X_train = [X1_train, X2_train]; % Combine the two arrays into one training array
model = TreeBagger(100, X_train, Y_train, 'Method', 'regression');

% Generate prediction grid
N = 101;
x = linspace(0, 1, N);
y = linspace(0, 1, N);
[X1p, X2p] = meshgrid(x, y);

% Make predictions on the grid
Zp = predict(model, [X1p(:), X2p(:)]);

% Reshape predictions to a grid
Z = reshape(Zp, N, N);

% Calculate predictions on the test set
Y_pred_test = model.predict([X1_test, X2_test]);

% Calculate evaluation metrics
mse = mean((Y_test - Y_pred_test).^2);
r2 = 1 - sum((Y_test - Y_pred_test).^2) / sum((Y_test - mean(Y_test)).^2);

% Plot the results with evaluation metrics
figure;
scatter3(X1_test, X2_test, Y_test, 'filled', 'MarkerFaceColor', 'b');
hold on;
surf(x, y, Z, 'FaceAlpha', 0.5);
xlabel('x');
ylabel('y');
zlabel('z');
title('Random Forest Model');
legend('Actual', 'Predicted', 'Location', 'Northwest');

% Display evaluation metrics on the graph
text(0.1, 0.8, sprintf('MSE: %.4f', mse), 'Units', 'normalized', 'FontSize', 12);
text(0.1, 0.75, sprintf('R2: %.4f', r2), 'Units', 'normalized', 'FontSize', 12);
```

A4.3 Code for decision tree model.

```
% Load the data from the CSV file
data = readmatrix('Data.csv');

% Split the data into input features (X) and target variable (Y)
X1 = data(:, 13);
X2 = data(:, 14);
Y = data(:, 16);

% Split the input features and target variable into training and test sets
rng(42); % Set a random seed for reproducibility
cv = cvpartition(size(data, 1), 'Holdout', 0.2); % 80% for training, 20% for testing
X1_train = X1(cv.training, :);
X2_train = X2(cv.training, :);
Y_train = Y(cv.training, :);
X1_test = X1(cv.test, :);
X2_test = X2(cv.test, :);
Y_test = Y(cv.test, :);

% Combine the training input features into one array
X_train = [X1_train, X2_train];

% Combine the test input features into one array
X_test = [X1_test, X2_test];

% Train a decision tree model
model = fitrtree(X_train, Y_train);

% Set the maximum depth for displaying nodes
maxDepth = 2;

% Create a copy of the tree with limited levels
limitedTree = prune(model, 'Level', maxDepth);

% View the limited-depth decision tree
view(limitedTree, 'Mode', 'graph');
```

A4.4 Code for ANN:

```
% Load the data from the CSV file
data = readtable('Data.csv');
X = table2array(data(:, 13:14));
Y = table2array(data(:, 16));

% Calculate column means
X_mean = mean(X, 'omitnan');
Y_mean = mean(Y, 'omitnan');

% Replace missing values with column means
X(isnan(X)) = X_mean;
Y(isnan(Y)) = Y_mean;

% Normalize the input variables
X_normalized = normalize(X);
X1 = X_normalized(:, 1); % Column 13 - Availability
X2 = X_normalized(:, 2); % Column 14 - Performance

% Define the neural network
net = fitnet(10);

% Set the training parameters
net.trainParam.epochs = 150000;
net.trainParam.goal = 0.000001;

% Train the neural network
net = train(net, [X1'; X2'], Y');

% Calculate evaluation metrics
Y_pred = net([X1'; X2']);
mse = mean((Y - Y_pred').^2, 'omitnan');
r2 = 1 - sum((Y - Y_pred').^2) / sum((Y - mean(Y)).^2);

% Create a table of evaluation metrics
evaluationTable = table(mse, r2, 'VariableNames', {'MSE', 'R2'});

% Display the evaluation table in the Windows prompt
disp(evaluationTable);

% Generate the surface plot
N = 101;
x = linspace(0, 1, N);
y = linspace(0, 1, N);
z = zeros(N, N);
for i = 1:N
    for j = 1:N
        input = [x(i); y(j)];
        output = net(input);
        z(j, i) = output;
    end
end
[Xmesh, Ymesh] = meshgrid(x, y);
Z = 100 * z;
figure;
surf(Xmesh, Ymesh, Z);
xlabel('X');
ylabel('Y');
zlabel('Z');
title('Surface Plot of Predicted Output');

% Generate the scatter plot of the input and output variables
Y_pred = net([X1'; X2']);
figure;
scatter3(X1, X2, Y, '.');
hold on;
scatter3(X1, X2, Y_pred', 'r');
xlabel('X1');
ylabel('X2');
zlabel('Y');
title('Scatter Plot of Input and Output Variables');
legend('Actual', 'Predicted');
```

A4.5 Code for comparison:

```
% Load the dataset
data = readtable('Data.csv');

% Extract the predictor variables (X) and target variable (y)
X = data{:, [13, 14]};
y = data{:, 16};

% Check if the dimensions of X and y match
if size(X, 1) ~= size(y, 1)
    error('The number of rows in X and y must be the same.');
end

% Remove rows with NaN values
nan_rows = any(ismissing([X y]), 2);
X = X(~nan_rows, :);
y = y(~nan_rows);

% Split the data into training and testing sets (e.g., 80% training, 20% testing)
rng(1); % For reproducibility
cv = cvpartition(size(X, 1), 'Holdout', 0.2);
X_train = X(cv.training, :);
y_train = y(cv.training);
X_test = X(cv.test, :);
y_test = y(cv.test);

% Train the models
degree = 3; % Degree for polynomial regression

% Polynomial regression
poly_coeffs = polyfit(X_train(:, 1), y_train, degree);
y_pred_poly = polyval(poly_coeffs, X_test(:, 1));

tree_model = fitrtree(X_train, y_train);
y_pred_tree = predict(tree_model, X_test);

forest_model = TreeBagger(50, X_train, y_train);
y_pred_forest = predict(forest_model, X_test);

ann_model = fitlm(X_train, y_train);
y_pred_ann = predict(ann_model, X_test);

% Convert y_pred_forest to numeric array
y_pred_forest = cellfun(@str2num, y_pred_forest);

% Calculate evaluation metrics: MSE, RMSE, and R2
mse_poly = mean((y_pred_poly - y_test).^2);
rmse_poly = sqrt(mse_poly);
mse_tree = mean((y_pred_tree - y_test).^2);
rmse_tree = sqrt(mse_tree);
mse_forest = mean((y_pred_forest - y_test).^2);
rmse_forest = sqrt(mse_forest);
mse_ann = mean((y_pred_ann - y_test).^2);
rmse_ann = sqrt(mse_ann);

r2_poly = 1 - sum((y_test - y_pred_poly).^2) / sum((y_test - mean(y_test)).^2);
r2_tree = 1 - sum((y_test - y_pred_tree).^2) / sum((y_test - mean(y_test)).^2);
r2_forest = 1 - sum((y_test - y_pred_forest).^2) / sum((y_test - mean(y_test)).^2);
r2_ann = 1 - sum((y_test - y_pred_ann).^2) / sum((y_test - mean(y_test)).^2);

% Create a table for the evaluation metrics
models = {'Polynomial Regression', 'Decision Tree', 'Random Forest', 'ANN'};
MSE = [mse_poly, mse_tree, mse_forest, mse_ann];
RMSE = [rmse_poly, rmse_tree, rmse_forest, rmse_ann];
R2 = [r2_poly, r2_tree, r2_forest, r2_ann];
evaluation_table = table(models', MSE', RMSE', R2', 'VariableNames', {'Model', 'MSE', 'RMSE', 'R2'});

% Display the evaluation table
disp(evaluation_table);
```