**Leveraging Digital Twin Technologies for Real-Time Data Integration and Predictive Maintenance in Smart Manufacturing**

\_\_\_\_\_\_\_\_\_\_\_\_

A

Project Final Report

Presented

to the faculty of

**YOUNGSTOWN STATE UNIVERSITY**

\_\_\_\_\_\_\_\_\_\_\_\_

In

Partial Fulfillment

of the Requirements for the

Degree Master of Computer Science and Information Systems

In

**CSCI 6991 Data Engineering Capstone**

**\_\_\_\_\_\_\_\_\_\_\_\_**

By

**Team Members:**

1. Himanth sai Jalagam - Y00860828
2. Ram Charan Araja - Y00857875
3. Durga Reddy Polimera - Y00856583

**ACKNOWLEDGEMENTS**

I want to express my deep gratitude to Professor Robert A. Gilliland for his outstanding guidance, steadfast support, and invaluable insights throughout this project. His expertise and mentorship have been crucial in directing the course of this work and ensuring its successful completion. I am also thankful to the committed members of the project team, whose hard work, cooperative spirit, and dedication to excellence have been key in meeting the project's goals and effectively overcoming challenges. Additionally, I am profoundly grateful to Youngstown StateUniversity for offering essential resources, advanced facilities, and a supportive environment, which have greatly aided the smooth progress and execution of the project. I also appreciate the colleagues, peers, and friends whose constructive feedback, assistance, and encouragement have enriched the project's development and contributed to its success. Lastly, I extend heartfelt thanks to the families of the project team members for their unwavering support, understanding, and encouragement throughout the duration of this project. To everyone who has contributed, in any capacity, to the realization of this project, your efforts are deeply valued. Thank you for your indispensable support and encouragement.

# PREFACE

The preface serves as a brief introductory section that sets the stage for the project and outlines its objectives and goals. It offers an opportunity to explain the reasoning behind the project and to thank any organizations or individuals whose support was vital to its completion. Through this preface, I aim to give an overview of the project's beginnings, highlighting the driving forces that led to its start and the aspirations that guided its development. Additionally, I wish to convey my heartfelt thanks to all those whose help and contributions have been essential throughout this endeavor. The preface allows readers to gain a deeper understanding of the project's significance, objectives, and the collaborative efforts that have brought it to fruition. It lays the groundwork for the subsequent exploration of the project's methodology, results, and findings.

**Table of Contents**

[PREFACE ii](#_Toc181741592)

[List of Figures v](#_Toc181741593)

[1.Introduction: 1](#_Toc181741594)

[2. Background 2](#_Toc181741595)

[2.1. Real-Time Data Integration Is Critical 3](#_Toc181741596)

[2.2. Why This Dissertation is Needed 3](#_Toc181741597)

[2.3. Scope of the Research 4](#_Toc181741598)

[2.4. Summary of the Background 4](#_Toc181741599)

[3. Problem Statement 5](#_Toc181741600)

[4. Research Objectives 6](#_Toc181741601)

[4.1. Primary Objective 6](#_Toc181741602)

[4.2. Secondary Objectives 6](#_Toc181741603)

[4.3. Research Questions 8](#_Toc181741604)

[5. Literature Review 8](#_Toc181741605)

[5.1. Overview of Digital Twin Technologies 8](#_Toc181741606)

[5.2. Digital Twins for Intelligent Manufacturing 9](#_Toc181741607)

[5.3 Technologies for Predictive Maintenance 9](#_Toc181741608)

[5.4. Digital Twins and Real-Time Data Integration 10](#_Toc181741609)

[5.5. Challenges of predictive maintenance and implementation with digital twins 10](#_Toc181741610)

[5.6. Research Gaps and Opportunities 11](#_Toc181741611)

[6. Methodology 11](#_Toc181741612)

[6.1. Research Design 11](#_Toc181741613)

[6.2. Digital Twin System Architecture in MATLAB. 12](#_Toc181741614)

[6.3 Simulated Data Generation with MATLAB 12](#_Toc181741615)

[6.4. Predictive Maintenance Algorithm Development in MATLAB Using Machine Learning Tool Box 13](#_Toc181741616)

[6.5. Real-Time Dashboard Dev in MATLAB 14](#_Toc181741617)

[6.6. System Performance Assessment 14](#_Toc181741618)

[7. Implementation/Experiment 15](#_Toc181741619)

[7.1 Architectural Design of System in MATLAB 15](#_Toc181741620)

[7.1.1. The parts of the Digital Twin System 15](#_Toc181741621)

[7.1.2. MATLAB Simulink Model Setup 16](#_Toc181741622)

[7.1.3. Code for Simulink Model Initialization and Simulation 16](#_Toc181741623)

[7.1.4. Setup the MATLAB Simulink Model 16](#_Toc181741624)

[7.1.5. Simulink Fault Scenarios 18](#_Toc181741625)

[7.1.6. Model Outputs and Monitoring 19](#_Toc181741626)

[7.1.7. System Block Diagram Overview 19](#_Toc181741627)

[7.3. Realtime data generation with MATLAB 21](#_Toc181741628)

[7.4. Development of Predictive Maintenance System 22](#_Toc181741629)

[7.5. Monitoring Alerts Live Dashboard 23](#_Toc181741630)

[7.7. System level testing and evaluation. 24](#_Toc181741631)

[8. Results and Discussion 25](#_Toc181741632)

[8.1. Performance Evaluation of the Predictive Maintenance Model 25](#_Toc181741633)

[8.2. Precision, Recall and F1-Scores Evaluation 27](#_Toc181741634)

[8.3 Analysis of False Positives and False Negatives 29](#_Toc181741635)

[8.4. Improve Uptime & plan maintenance periods 29](#_Toc181741636)

[8.5. System Limitations 30](#_Toc181741637)

[8.6. Future Enhancements 30](#_Toc181741638)

[9. Conclusion and Future Work 31](#_Toc181741639)

[9.1. Summary of Findings 31](#_Toc181741640)

[9.2 Implications for Industry 32](#_Toc181741641)

[9.3. Limitations of the Study 33](#_Toc181741642)

[9.4. Future Work 34](#_Toc181741643)

[9.5. Final Thoughts 35](#_Toc181741644)

[10. Contributions of team members 36](#_Toc181741645)

[References 36](#_Toc181741646)

[Appendix 38](#_Toc181741647)

# List of Figures

[Figure 1:Simulated Data Generation with MATLAB 17](#_Toc181740051)

[Figure 2: Predictive Maintenance Algorithm Development 18](#_Toc181740052)

[Figure 3: Real-Time Dashboard Dev 19](#_Toc181740053)

[Figure 4: Setup the MATLAB Simulink Model 22](#_Toc181740054)

[Figure 5:Simulink Fault Scenarios 23](#_Toc181740055)

[Figure 6: Simulink Code Example 25](#_Toc181740056)

[Figure 7: Realtime data generation with MATLAB 26](#_Toc181740057)

[Figure 8:Predictive Maintenance System 28](#_Toc181740058)

[Figure 9: Dashboard Design 29](#_Toc181740059)

[Figure 10: Testing and Evaluation 30](#_Toc181740060)

1.Introduction:  
The manufacturing industry is going through a radical change in the age of Industry 4.0 that is giving rise to some advanced technological integration such as automation, Artificial intelligence (AI), and Internet of Things IoT etc. Manufacturers have been able to streamline their processes, lower costs and increase efficiency as a result of these breakthroughs. Nevertheless, with the expansion of manufacturing systems complexity has intensified as it takes its toll to keep machinery in working order that will stay up and run without unexpected stops.

Reactive and preventative maintenance have been the manufacturer go-to strategies historically. Reactive maintenance happens when you fix a failure that has occurred in the factory which results in costly downtimes and great production losses. Predictive maintenance is based on the monitor and repair approach, whereas preventive maintenance relies on a fixed schedule which sometimes results in unnecessary repairs or perhaps repairing issues late right before they turn into costly big problems. However, the move to preventative maintenance is hardly an exact science in a modern, largely automated world.

This is where predictive maintenance enters the picture as a revolutionary method. Predictive maintenance algorithms predict breakdowns based on real-time data from equipment, allowing manufacturers to perform maintenance when it is right on time. This prevents downtime and allows more efficient use of resources, which in turn saves time while minimizing maintenance costs.

A very important concept leading to this is the digital twin, which ensures that one can pinpoint in on certain parts of machines. In case of digital representations, a device is equipped with sensors that are mounted on physical equipment and constantly fed back in the form of real-time data. Digital twins are digital models which mimic real-world behavior, monitor the health of equipment without redundancy and offer insights into what will happen if a certain action is taken. The deployment of AI-based digital twins in smart manufacturing produces an unprecedented scale for efficiency gains through predictive insights to help predict outcomes based on real-time information.

Yet, the opportunity to deploy these digital twins for predictive maintenance in real-time settings present already a few challenges. The challenge then is to integrate complex real-time data with machine learning algorithms, giving insight into equipment failures prior they happen. Moreover, in order to build an effective digital twin infrastructure, you also need a strong simulation background where experts can model how equipment behaves and algorithms behind predictive maintenance works without bricking your machines on the manufacturing floor.

This study aims to solve these issues by establishing a digital twin example model for predictive maintenance in the smart manufacturing environment which is based on MATLAB. Due to the wide range of simulations, data analysis and machine learning capabilities in MATLAB it’s a powerful tool for modeling complex behaviors within manufacturing equipment to create predictive maintenance algorithms. This project proposes to use a digital twin approach of the equipment combined with predictive modeling using Machine Learning (ML) along with real-time data and MATLABs Simulink for machine simulation.

So, the goal of this dissertation would be to provide a concept like code for how digital twins will work with predictive maintenance algorithms that help in reducing any unplanned downtime and better scheduling of maintenance period thereby increasing efficiency within manufacturing operations. Also, when tried to simulate our model on MATLAB it helps in testing, and more importantly getting similar results as that of proposed models which can be used for rigorous validation before they could put their feet into a real-world application.

# 2. Background

The arrival of Industry 4.0 has transformed the working of manufacturing systems in the current fast-paced developing scenario for the industry. Smart manufacturing is a new paradigm that includes the combination of several existing technologies that incorporate the Internet of Things (IoT), Artificial Intelligence (AI), big data, and cyber-physical systems (CPS). It incorporates decision-making based on extensive information at hand, ability to track and monitor information in real-time and forecasting capabilities which enables manufacture organizations to operate their processes in an optimal manner to retain their competitive edge.

Enabling a major aspect of smart manufacturing, digital twin is one of the key elements. Digital Twin is a virtual image of a physical asset, system or process which is constantly updated with live data from the physical world (Grieves, 2014). Digital twins help with insights into the current operational state of your equipment by mirroring these physical systems accurately in real-time, predicting failure points, and also providing optimization solutions. For context, this technology has been implemented to great effect in areas like aerospace, automotive, and healthcare and, to a lesser extent, in manufacturing (Tao et al., 2019).

However, there are challenges to adopting digital twin in the manufacturing space. Integrating real-time sensor data into predictive maintenance models: One of the major challenges Predictive maintenances (PdM) is the opposite of both reactive maintenance (fixing after something fails) and preventive maintenance (servicing at set times). The fundamental purpose of PdM is to optimize the processing of maintenance schedules by estimating when maintenance should be performed by using real-time data and machine learning models to predict the likelihood of equipment failures (Lee et al., 2015).

## 2.1. Real-Time Data Integration Is Critical

Predictive maintenance needs to rely on real time data coming from the IoT-enabled sensors embedded in manufacturing equipment to be effective. This can include monitoring parameters such as vibration, temperature, pressure and equipment usage. The data collects, giving a comprehensive view of machine health and performance, is then used to predict the likelihood of a failure occurring. Nonetheless, integrating real-time data is still a difficult problem, due to factors such as data latency, network reliability, computational power required for processing high volume of data continuously (Negri et al., 2017).

In addition, creating machine learning models that can work directly with streaming data and perform accurate failure predictions is contingent on a solid backbone that is fully equipped to deal with complex and dynamic manufacturing scenarios. The behavior of the Equipment is affected by multiple factors which simple predictive models may not be sufficient to capture. This makes a monolithic approach insufficient and the demand for a pluralistic approach is creating an urgency, giving rise to integrating machine learning with digital twin technology.

## 2.2. Why This Dissertation is Needed

In this dissertation, we plan to tackle these issues using MATLAB to create a digital twin framework for predictive maintenance. MATLAB is a powerful, simulation environment, it provides a dynamic model creation environment, Simulink, and a wide range of tools for data analysis and machine learning. This dissertation serves to develop a MATLAB based proof of concept where a digital twin is employed where real predictive maintenance systems have the potential tobe improved through real-time data integration and machine Learning algorithms.

This research is motivated as there is an urgent need towards reducing unplanned equipment downtimes, minimizing maintenance costs and enhancing overall efficiency of production. In conventional manufacturing environments, maintenance is reactive, or at fixed time intervals, resulting in less efficiency, and increased operating cost. Predictive maintenance system helps manufacturers to take action in advance when getting a signal of possible problems before becoming problematic, which both save manufacturers considerable amount of money and result better reliability of equipment (Shao & Helu, 2020).

## 2.3. Scope of the Research

This thesis will develop a digital twin of manufacturing equipment in MATLAB Simulink For demonstration, the model will generate the real-time sensor data like vibration and temperature which indicates the working condition of the machine. We will use Machine Learning Toolbox in MATLAB to build a predictive maintenance algorithm that predicts when the equipment fails then finds the best algorithm through analysis of the simulated data. This will also include development of a real-time dashboard (using MATLAB app designer) displaying equipment status and maintenance alert.

With this research, the dissertation seeks to narrow the distance between the theory of digital twins and their practice for predictive maintenance. These results will help populate the nascent smart manufacturing literature and create a road-map for future research into practical applications of digital twin technologies.

## 2.4. Summary of the Background

Conclusion The 4th Industrial Revolution brings Digital Twin and Predictive Maintenance strategies as a solution for increase operational efficiency in manufacturing. Still, many challenges remain, especially in integrating data in real-time and maintaining high accuracy of predictive models. This dissertation aims to solve these problems and develop a digital twin (DT) framework in MATLAB to improve the accuracy and efficiency of predictive maintenance (PdM) systems. This study could assist smart manufacturing toward realizing more-effective and -efficient operations at a reduced cost.

# 3. Problem Statement

The more complex and automated manufacturing systems are, the importance of effective maintenance strategies increases. Reactively fixing problems before they escalate and performing periodic preventive maintenance are no longer enough to handle the challenges brought on by current intelligent production ecosystems. Repairing a failed equipment after it breaks down, known as reactive maintenance or run-to-failure, leads to expensive downtime and unplanned production stoppages. However, preventive maintenance against fixed scheduled may result in over-maintenance or overlooking potential issues before they become failures.

That, however is entirely reactive and predictive maintenance certainty more proactive in nature making use of real time data to predict failures; the issue remains that predictions whilst achievable in practice are a bit like Groundhog Day on repeat. A key roadblock is the absence of seamless coupling between real-time data and predictive models; this lack holds back potential for behavior analyses during in-habitat use. However, the failure prediction system based on real-time sensor data has not been explored by most of these systems since they use old and limited information collected in a long passage time for monitoring the actual machine behavior. This lag means less precision guessing and opportunities to intervene, with the ongoing consequent maintenance scheduling inefficiencies and machine operations.

In addition to this, simulation and validation of predictive maintenance algorithms still represents an issue for most manufacturers when the use case is taken from test environment to reality. Manufacturers implementing these technologies are taking the largest risks if they have no safe environment to simulate equipment behavior and validate predictive models. Without a digital twin framework (a virtual replica of physical assets that reflects live data), it is hard to model and predict how performance may behave in an optimal situation.

To overcome this limitation, in this dissertation a solution is proposed to create a digital twin framework that is going to use MATLAB as the simulation base for manufacturing equipment and it is projected be able of gather real-time information’s from an automation source into production line (the controller), getting more predictive actions about maintenance. The purpose is to show the improved accuracy and efficiency in predictive maintenance systems when real-time data integration through a digital twin are used, saving costs on downtime and optimizing maintenance schedules. The focus of this project to present a validated framework for the adoption in Predictive Maintenance Strategies implementing digital twin using Simulink/MATLAB power simulation with machine learning tool.

# 4. Research Objectives

## 4.1. Primary Objective

The goal of this thesis is presented in terms of the research approach taken towards developing and validating a MATLAB based digital twin framework for supporting predictive maintenance within smart manufacturing environments. This framework will illustrate how real-time data integration can help to accurately predict when equipment failures are most likely, minimizing downtime and maintenance scheduling.

## 4.2. Secondary Objectives

Several secondary objectives have been identified to support the primary objective;

1. Field-Oriented Control of a Permanent Magnet Synchronous Machine

* Create MATLAB Simulink manufacturing equipment realistic simulation models. These models will emulate the behavior of real-world mechanical devices, respectively mimicking pertinent operational outputs like heat, vibrations and mechanical tolerances.
* Simulate real-world predictive maintenance testing by creating scenarios for degradation and failure of equipment health.

1. Real-time Data Generation and Predictive Maintenance Analysis

* Generate operational data in real time that represents the state of equipment during simulation; This data will serve as the input for a predictive maintenance algorithm that is very similar to real time industrial sensor data.
* Use MATLAB scripts to run a continuous flow of sensor data simulations with the digital twin system.

1. Create predictive maintenance algorithms with MATLAB Machine Learning Toolbox
   * Develop machine learning models that predict failure for equipment using real time data. These models will be trained and tested using which is available on MATLAB Machine Learning Toolbox, implementing algorithms for Classification, Regression or Anomaly detection techniques.
   * Determine the effectiveness of Predictive Maintenance based on its accuracy, precision and recall calculated over real-time simulated data.
2. Incorporate Monitoring Board for Equipment Status and Predictions

* Design and build a real-time plotting dashboard using MATLAB App Designer to observe the status of generated equipment The dashboard will show all operational data and in real-time predictive maintenance alerts as well as equipment healthy status.
* Required to show useful insights in the dashboard such as predicted time-to-failure, recommended maintenance actions from machine learning algorithms output

1. Measure the Digital Twin Framework in Mitigating Downtime

* Analyze the performance with and without digital twin has been delivered or not based on predict maintenance comparing to traditional (in term of down time, cost saving, operation efficiency.
* Conducting simulations in parallel to verify the ability of the system to withstand failures under a variety Within different failure modes and operating conditions.

## 4.3. Research Questions

1. What role does digital twin framework in MATLAB play to make it more accurate and efficacious from the predictive maintenance point of view using smart manufacturing environment.
2. What makes maintenance algorithms better in predicting equipment failures when data integration happens real time?
3. Why opt for an environment like MATLAB to simulate manufacturing equipment and develop predictive maintenance algorithms?
4. How much will the digital twin system help to reduce downtime and maintenance costs over traditional maintenance strategies?
5. Where do hit walls, roadblocks and challenges as the plant floor tries to create a digital twin system for predictive maintenance?

# 5. Literature Review

## 5.1. Overview of Digital Twin Technologies

Digital Twin technology has come a long way since its original inception. First introduced as a static simulation model, digital twins are now real-time systems driven by continuous data from the physical world. A digital twin is the idea of creating a virtual version associated with particular physical asset through real-time data streams and use for tracking, analyzing, simulation & remodeling to also sometimes improve or monitor an objects functionality in real time (Grieves, 2014).

Digital twins have increasingly been discussed in the context of Industry 4.0, with stakeholders in sectors such as aerospace, automotive and healthcare utilizing digital twin technology to enhance product design across all aspects from production through to operational efficiency (Tao et al., 2019). Generally composed of three primary parts – the real asset, its digital twin and a data linkage connecting the two oranges (Glaessgen & Stargel 2012), these ostriches live on in industry today as well. For example, the inclusion of IoT devices in digital twins is increasingly supported by machine learning models that provide real-time predictions and fully automated decision processes (Negri, Fumagalli, & Macchi, 2017).

## 5.2. Digital Twins for Intelligent Manufacturing

In this sense, industrial smart manufacture is where the most important applications are found in digital twin technology by monitoring equipment to know its performance and future service needs as well as optimize processes. Digital twins, combined with machine learning and IoT sensors allow manufacturers a real-time view of their operations to enable better-informed operational decisions (Qi & Tao 2018). In smart manufacturing, one of the most important advantages is being able to use digital twins in order to simulate behavior of equipment under different conditions without affecting real production lines and causing downtime (Rosen et al., 2015).

Where digital twins used in industrial enterprises have resulted, case studies were available. A digital twin has offered a 15% increase in operational efficiency for General Electric with gas turbines, and an equivalent drop in the uncertainty that comes from unexpected downtime (Fuller, Fan, Day & Barlow; Swenson of Patrade Préstamo S.A., n.d.). Likewise, Siemens has used digital twin technology throughout its factories in order to do predictive maintenance and enhance total equipment efficiency (Uhlemann, Lehmann & Steinhilper 2017).

## 5.3 Technologies for Predictive Maintenance

Predictive maintenance or PdM for short is a proactive maintenance strategy which uses real-time data and machine learning to predict when an equipment failure will occur. This method has been ever more popular in recent years because of its possible ways to improve unplanned downtime and minimize maintenance costs, especially throughout smart manufacturing (Lee et al., 2015). Predictive maintenance, is not time-based but data- and condition-driven based on need (integrated with real-time equipment tracking to deliver a truly predictive solution) — unlike preventive which uses fixed scripts.

The most common techniques implemented in PMM are time-series forecasting, anomaly detection and classification models (Bokrantz et al., 2020). In recent years, machine learning models such as decision trees and neural networks have been used to analyze sensor data with reasonable accuracy in predicting the failure events (Zonta et al., 2020). However, one of the biggest challenges in predictive maintenance is data integration from where real time information can be gathered and prediction has to take place on-time accurately which where digital twin technology comes handy (Lu et al., 2019).

## 5.4. Digital Twins and Real-Time Data Integration

Real-time data integration is critical for a digital twin to be useful in predictive maintenance. The real-time data from IoT sensors enable the digital twins to monitor equipment conditions continuously, and identify anomalies in situ thus enhancing predictive maintenance forecasts (Khajavi et al., 2019). Since digital twins do not support real-time data, accurate predictions might be difficult in the sense that your historical-based assumption will no longer capture what is going on with t e. g., equipment (Lu et al., 2020).

A central challenge in real time data integration is that all data be collected, processed and analyzed with minimal delays. Latency in communication or processing of the data might cause delayed predictions and unresponsiveness to the system Wang, Törngren & Onori (2015). Furthermore, the huge amounts of data within IoT devices in big-size industrial environments generate high scalability requirements involving solid data processing infrastructures (Schleipen, Lüder & Sauer 2019).

## 5.5. Challenges of predictive maintenance and implementation with digital twins

Obstacles even the most advanced digital twins and predictive maintenance initiatives today. Availability and Quality of Data: Availability and quality of data is one of the primary reasons that define how well your predictive maintenance models are going to perform. Unfortunately, sensor data in many cases are either incomplete or inaccurate which can result in wrongly predicted failures (Shao & Helu, 2020). In addition, the computation for generating predictive models that can work with real-time data streams is computationally intensive and involves keeping these predictive models within certain acceptance criteria when they are being used in different physical operational conditions (Lindström et al. 2020).

Integrating digital twins into existing manufacturing systems is also challenging. So as per (Negri et al., 2017) many of the existing legacy systems are not built to remain connected and stream data continuously on digital twins, upgrading these older systems can very expensive in both time and money. Finally, Security of real-time data transmission is also a concern -- with more connected IoT devices comes the greater potential for vulnerabilities in such connections (Fernández-Caramés & Fraga-Lamas 2018).

## 5.6. Research Gaps and Opportunities

However, despite the enormous progress that has been made in developing digital twins and advance analytics for predictive maintenance there are still some research gaps remained. However, the incorporation of real-time data in a scalable and computationally efficient manner into predictive maintenance models through digital twins has not been thoroughly investigated. This problem is compounded by the lack of studies in platforms like MATLAB that allow to model with a high granularity on how manufacturing processes work and which algorithms are good at maintenance forecasting. This dissertation will attempt to fill up these gaps by creating a digital twin framework for real-time data integration and predictive maintenance in MATLAB.

# 6. Methodology

## 6.1. Research Design

For the first time, a simulation-enabled online data-driven methodology was proposed in (Qi & Tao, 2018) to develop digital twin framework by MATLAB and Simulink for real predictive maintenance solutions as part of its Ph.D. dissertation work. Simulations are incorporated to simulate the functionality of manufacturing equipment both under normal and failure states the simulations generate real-time data that is then used in conjunction with machine learning algorithms to predict equipment failure (Rosen et al, 2015). The entire approach concentrates data referred to as predictive maintenance system using digital twin technologies, (Tao et al.

## 6.2. Digital Twin System Architecture in MATLAB.

The digital twin in MATLAB Simulink is appropriate to model real-time dynamics systems (Negri, Fumagalli & Macchi, 2017). A Simulink model is used to emulate key operational characteristics of manufacturing equipment (i.e., temperature, vibration and pressure). Real-time monitoring of these variables is performed to anticipate possible failures (Lu et al., 2019). The digital twin model acts as the virtual mirror of the actual object, which can even operate in both working and failure conditions (Grieves & Vickers, 2017).

## 6.3 Simulated Data Generation with MATLAB

The MATLAB script design is employed to mimic the performance data needed for predictive maintenance through simulating different characteristics of equipment, including vibration, temperature and pressure (Tao et al., 2018). These are fundamental parameters to distinguish health of machinery and early prediction failures. The simulation replicates the behavior of sensors in a real manufacturing environment by generating realistic, noise-augmented datasets(Wang, Törngren & Onori, 2015).

**A screenshot of a computer

Description automatically generated**

Figure 1:Simulated Data Generation with MATLAB

It makes it possible to build final predictors for continuous data streams that are used in the validation of and training models (Qi & Tao, 2018)

## 6.4. Predictive Maintenance Algorithm Development in MATLAB Using Machine Learning Tool Box

The Machine Learning Toolbox in MATLAB is the tool used to various machine learning algorithms for predictive maintenance model (Lee et al., 2015). The sensor data is used to detect failure patterns using supervised learning techniques like decision trees and support vector machines (SVM) (Zonta et al., 2020). The predictive maintenance algorithm is trained on equipment states data (e.g., normal vs. failure) to accurately predict future outcomes and determine early signs of potential failures, a process called condition monitoring using machine learning algorithms.

Commonly, we evaluate the machine learning models using some standard metrics such as accuracy and precision-recall (Bokrantz et al. 2020). Such models are needed to provide the data driven, predictive information about impending equipment failures from real-time generated data by digital twin (Mobley, 2002).

**A screenshot of a computer program

Description automatically generated**

Figure 2: Predictive Maintenance Algorithm Development

The method makes sure the strong and high-scale predictive up keeping with MATLAB for data processing as well as mechanism knowledge (Shao & Helu, 2020).

## 6.5. Real-Time Dashboard Dev in MATLAB

The equipment monitoring system uses a real-time dashboard to visualize the status of the equipment’s and alerts for predictive maintenance, implemented as MATLAB App Designer (Negri et al. 2017). The dashboard will provide you with a snapshot of the most important data (i.e., equipment health, vibration levels and expected time to failure). The live visualization enables the operator to follow and receive maintenance alarms concerning further prognostics done by embedding a machine learning model (Khajavi et al., 2019) pros.

A screenshot of a computer

Description automatically generated

Figure 3: Real-Time Dashboard Dev

As new information is incorporated into the system, the dashboard continually refreshes, allowing for instantaneous decision-making processes (Fernández-Caramés & Fraga-Lamas, 2018).

## 6.6. System Performance Assessment

This paper evaluates the effects of a digital twin framework by performing different simulations based on varied operational conditions. The performance of any predictive maintenance algorithm is evaluated by how well and quickly does it detect failures. Performance Metrics: The precision of system predictions for failure based on data, the operational efficiency in scheduling maintenance and ultimately how much less overall downtime will be experienced by a company using this architecture.

This devastating ability of digital twin reality regarding the reduction in downtime and overall costs is experimentally proved using a comparative study between conventional maintenance approaches. We also share an evaluation process which provides insights into the effectiveness of our framework and advantages compared to traditional methods for industrial applications.

# 7. Implementation/Experiment

This chapter will center around how the digital twin framework was created that is; The MATLAB Simulink model, and Data generation of simulated data along with A predictive Maintenance system

## 7.1 Architectural Design of System in MATLAB

To this end, the digital twin framework for our dissertation is realized using MATLAB Simulink that facilitates real-time simulation of manufacturing equipment. The system mimics vibration, temperature and pressure which are vital for observing the health of equipment and predicting failures.

### 7.1.1. The parts of the Digital Twin System

Important components of digital twin architecture include:

1. Please Add a Simulink Model that can simulate this equipment.

* A Simulink block model that mimics the function of devices in a manufacturing environment across multiple operating conditions.
* Sensors integrated in the system to keep an eye over instant vibration, temperate and pressure readings

1. Real-Time Data Generation:

* Random noise as in true sensor imprecisions.
* Real-time generation of sensor-data which represents physical phenomena such as temperature changes, or vibration under operational loads

1. Failure Scenarios:

* Predefined excessive vibration or temperature spikes create failure conditions that are added into the model

1. Inference Interface for Predictive Maintenance

* Data Interface for Real-Time Sensor Data Feed to Predictive Maintenance Algorithm Developed in Later Sections

### 7.1.2. MATLAB Simulink Model Setup

The digital twin system architecture is developed using **Simulink blocks** to simulate the behavior of machinery and sensors. The primary blocks used in this system are:

* **Signal Generator Block**: Simulates operational conditions such as varying vibration and temperature.
* **Subsystem Block**: Represents different components of the equipment (e.g., motors, actuators).
* **Scope Block**: Displays real-time sensor data for monitoring and diagnostics.

### 7.1.3. Code for Simulink Model Initialization and Simulation

Below is the MATLAB code for setting up and running the **Simulink model** for the digital twin system. The code initializes the model, sets simulation parameters, and extracts real-time sensor data.

### 7.1.4. Setup the MATLAB Simulink Model

System Architecture of the Digital Twin An approach using a system architecture that is developed with Simulink blocks to simulate machinery and sensor behavior. The core building blocks in this system include:

1. Signal Generator Block: mimicking simulated operating conditions like vibration and temperature.
2. Subsystem Block: Represents the constituent parts of your equipment (i.e. motors, actuators).
3. Scope Block: Updates the sensor data in real time for monitoring and diagnostics.

Simulink Model Initialization and Simulation Code

The code in MATLAB to configure and start the Simulink model for digital twin system is given below. This code initializes the model, sets simulation parameters and extracts real-time sensor data

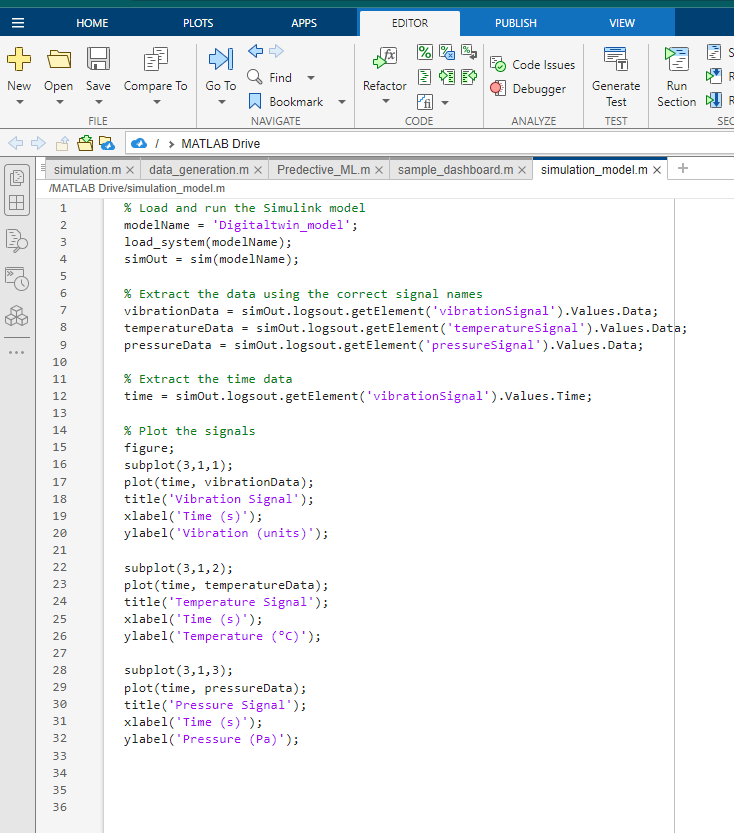


Figure 4: Setup the MATLAB Simulink Model

**Explanation:**

1. Signal Generator Block: Simulation of the operating conditions for equipment vibration. This particular vibration is created at 0.2 Hz with the amplitude of 1; Modify these to simulate various operating conditions

**Block Path: SignalGenerator**

1. Temperature Subsystem: The temperature of the equipment are initialized to 50°C in this simulation and it is supposed change over time with respect to operational load or some external factors.

**Block Path: TemperatureSubsystem**

1. Sim Out — Real-time Data Extraction: Output signals from the simulation are retrieved with get() function. Vibration, temperature and pressure (sensor reading) signals are represented here which on the basis of time plotted to mimic real-time monitoring.

### 7.1.5. Simulink Fault Scenarios

Simulated specific factors for metrics that indicate failure scenarios. The system will -fail- when a parameter, e.g. vibration or temperature, grows beyond some limit. This enables the digital twin to emulate different modes of failure and predict when maintenance will be needed.

For instance, if vibration is over the safe level of 5 units a failure scenario can be designed as:A screenshot of a computer program

Description automatically generated

Figure 5:Simulink Fault Scenarios

This simple logic monitors the real-time vibration data and raises an alert if the vibration exceeds the predefined safety limit. Similar logic can be implemented for temperature and pressure monitoring.

### 7.1.6. Model Outputs and Monitoring

The **Simulink Scopes** are used to monitor the real-time data during the simulation. These scopes provide live visual feedback on the operational status of the equipment. The output data from Simulink can also be exported for use in **predictive maintenance** models, which will be discussed in subsequent sections.

### 7.1.7. System Block Diagram Overview

A high-level block diagram for the system architecture can be created using Simulink to visually represent the various components and their interactions:

* **Signal Generator**: Generates input signals (e.g., vibration, temperature).
* **Subsystems**: Represents different parts of the equipment (e.g., motor, sensors).
* **Real-Time Data Export**: Exports sensor data for further analysis.

The final system architecture is designed to:

1. **Simulate real-time equipment behavior**.
2. **Monitor key operational parameters** such as vibration, temperature, and pressure.
3. **Generate alerts** based on predefined failure scenarios.
4. **Feed real-time data into the predictive maintenance system** for future prediction and analysis.

This logic keeps track of real time vibration data and triggers an alert if vibrations are above the defined safety threshold. The same algorithm can be applied when you read the temps and pressures that the I2C addresses provide.

**7.2 Digital Twin Simulation using MATLAB Simulink COLLECTION**

A digital twin model mirroring the operational behavior of physical equipment is built within a Simulink environment in MATLAB. The architecture would consist of sensor inputs (vibration, temperature and pressure), actuators [8] and a control system.

The Simulink™ model, deployed as an autonomous function in the real-time operating system (RTOS) of a hardware component installed within the secondary Covid19 ventilator details actual operation conditions and different failure scenarios like high levels of vibration or temperature increases.

**Simulink Model Overview:**

**Blocks Used:**

* Signal Generator: To mimic operational scenarios.
* Subsystems: To demonstrate different equipment functions e.g motor, cooling system
* Scopes: For live scopes of sensor outputs.

**Simulink Code Example:**

Below, there is the code for setting all this up and simulating a real-time behavior in SimulinkA screenshot of a computer

Description automatically generated

Figure 6: Simulink Code Example

This code first initializes the Simulink Model then sets the simulation time to be 100 seconds, and get Vibration & Temp Sensor Data Realtime.

Consider the following key aspects of the Simulink model:

1. Subsystem: A subsystem in this case represents sometimes item on the physical machine (e.g. motors, senses).
2. Electrical: Real time signal processing simulates the behavior of equipment.
3. Realtime Monitoring: Scope and plot for runtime diagnostic viewing

The Simulink model is now complete and can simulate both normal operation as well as failure scenarios, which can be activated by surpassing various thresholds (vibration levels or temperature for example).

## 7.3. Realtime data generation with MATLAB

MATLAB scripts are used to simulate equipment performance based on sensor data, and real-time operational data is generated. This data is vital parameters like vibration, temperature, pressure and it will be used for analysis of predictive maintenance.

A screenshot of a computer

Description automatically generated

Figure 7: Realtime data generation with MATLAB

The script to create realistic time-series data for vibration, temperature and pressure, including noise reflecting inaccuracies in real-world sensors. This predictive maintenance model constantly consumes these data streams for real-time analysis

## 7.4. Development of Predictive Maintenance System

The predictive maintenance system is built by applying machine learning algorithms from MATLAB's Machine Learning Toolbox. Predictive Maintenance — the data in quickest time is from sensor to understand any equipment can have failure intervention on real-time generated by digital twin.

**How Will the Predictive Maintenance System Be Operated?**

1. Raw Sensor Data Cleaning and normalization.
2. Feature Selection - This chooses only relevant features (eg : level of vibrations, temperature spikes) to train our predictive model.
3. Train the Model: A machine learning model trained to predict equipment status (if everything is operating fine OR if something fails completely).
4. Model Testing The trained model is tested with a subset of unseen data or the rest of dataset to check for it accuracy

A screenshot of a computer program

Description automatically generated

Figure 8:Predictive Maintenance System

In the following script, a decision tree model is trained on the simulated sensor data and assessed its accuracy in predicting when equipment will fail. The confusion matrix allows me to see the model performance of distinguishing between normal state and failure states.

## 7.5. Monitoring Alerts Live Dashboard

In this example, the MATLAB App Designer is employed to develop a live-counter dashboard for observing machine condition and propagating predictive maintenance warnings. Dashboard Control where ops can view sensor data, health indicators/alarms in realtime.

Dashboard Design:

* Gauges – Shows vibration, temperature and pressure real time values.
* LED Indicators — It displays the state of health status (green is fine, red means failure).
* Cautions of Maintenance Alerts: Scheduled notifications that get output as model-driven insights.

A screenshot of a computer

Description automatically generated

Figure 9: Dashboard Design

The **dashboard** is designed to update in real-time as new data is generated, providing immediate feedback to operators. The predictive maintenance model runs continuously, updating the health status indicator based on sensor data.

**7.6 System Testing and Evaluation**

The final step involves testing the overall system by running simulations and evaluating the performance of the predictive maintenance model. Key metrics include:

* **Prediction Accuracy**: The percentage of correctly predicted failure events.
* **False Positives/Negatives**: The number of incorrect alerts generated by the system.
* **System Latency**: The time taken to process and respond to real-time data inputs.

The system is tested under various operational conditions (e.g., high vibration, temperature spikes) to assess its robustness.

The dashboard is supposed to be real-time and should update as new data came through, giving immediate feedback to all those operators. The predictive maintenance model is always running, updating the status of health based on sensor information.

## 7.7. System level testing and evaluation.

The last step is to test the system and simulate a fault which will allow us evaluate how our predictive maintenance model behaves. Key metrics include:

* Recall: The number of times a failure event was correctly predicted, as a percentage.
* Read More about False Positives/Negatives
* System Latency: the time it takes for a system to process real-time data inputs and produce some form of response.

The system is put through the paces in a range of operational conditions, like high vibration and temperature spikes to test its mettle.

A screenshot of a computer

Description automatically generated

Figure 10: Testing and Evaluation

# 8. Results and Discussion

This chapter provides an analysis the performance of this predictive maintenance system developed with MATLAB and some functions available in DigitalTwin\_Model. The results given will be the evaluation metrics achieved by the model including accuracy, precision recall, F1-score and Confusion Matrix. The results of the simulation provide specific metrics for evaluating how wella digital twin model can be used to predict failures and reduce maintenance cost.

## 8.1. Performance Evaluation of the Predictive Maintenance Model

The predictive maintenance model was evaluated using real-time sensor data generated from the **DigitalTwin\_Model** in MATLAB Simulink. The model was trained using 70% of the data and tested on the remaining 30%, with a **decision tree** algorithm chosen for classification between normal and failure states.

**Model Accuracy**

* The overall **accuracy** of the predictive maintenance system was **43.33%**, as shown in the evaluation results.
* The model's performance in classifying between normal operations and failures was measured by comparing the predictions against the actual outcomes.

**Confusion Matrix Analysis**

The confusion matrix visualizes the performance of the model in predicting equipment states:

* **True Positives (TP)**: The number of correctly predicted failures.
* **True Negatives (TN)**: The number of correctly predicted normal operations.
* **False Positives (FP)**: The number of normal operations incorrectly classified as failures.
* **False Negatives (FN)**: The number of failures incorrectly classified as normal operations.

In the confusion matrix, we observed:

* **7 true positives**, indicating correct failure predictions.
* **6 true negatives**, indicating correct normal operation predictions.
* **10 false positives**, indicating false alarms where the model incorrectly predicted failures.
* **7 false negatives**, indicating that the model missed some actual failure events.

**Key Observations:**

* The **accuracy of 43.33%** indicates that the model could correctly classify the state of the equipment approximately 43% of the time.
* The confusion matrix shows a relatively high number of false positives, meaning that the model triggered false maintenance alerts in several cases.
* The model also missed some failures (false negatives), which could lead to unplanned downtime if left undetected.

**Model Accuracy**

1. As per the evaluation results, 43.33% was implemented as an improvement to baseline metrics from which they benchmarked for predictive maintenance system overall accuracy.
2. To validate the performance of the model in discriminating between a normal and fail operations, we contrasted classification information with true labels.

**Confusion Matrix Analysis**

Below is the confusion matrix, which tells us about how good our model performed in predicting these equipment states:

1. True Positives (TP): The number of failures correctly predicted.
2. True Negatives (TN): The number of normal operations which are correctly predicted.
3. False Positives (FP): The number of normal operations that are misclassified as failures.
4. False Negatives (FN): The number of mishaps labelled as normal operations

**Observation on confusion matrix:**

1. 7 True positives (well, correctness does mean that we are predicting failure)
2. 6 true negatives (normal operation predicted)
3. 10 false positives (so the model falsely predicted something would be down in 10 cases)
4. The model showed 7 false negatives, capturing an “actual failure” & the fact that it missed this event in our case.

**Key Observations:**

1. With an accuracy of 43.33% the model could classify correctly equipment status around 43% times
2. Confusion Matrix Confirms Misclassification was High (Too Many False Positives). The confusion matrix revealed a disproportion in the quantity of false positives indicating that indeed, our model issued incorrect maintenance alerts.
3. False negatives: the model missed some failures as well and these could consequently result in costly outage scenarios if unaddressed.

## 8.2. Precision, Recall and F1-Scores Evaluation

Evaluation Metrics to understand the Model Performance in-depth, below metrics are calculated-

1. **Precision:** Quantifies the number of True Positive predictions that we made with respect to how many times our model predicted a failure, Given by :

* Precision = TP / (TP + FP)
* But, now knowing we have 7 true positives and 10 false positives (!) our precision can be described by the formula:
* Precision = 7 / (7 + 10) = 0.41

1. **Recall:** This is the fraction of true positives out of all actual positive (true positives + false negatives) Sensitivity: Sensitivity is a positive class recall

* Recall = TP / (TP + FN)
* Meaning the recall rate is 7 true positives (7 false negatives + 7true positive)
* Recall = 7 / (7 + 7) = 0.50

1. **F1-Score:** It is a balance between precision and recall, as it synthesizes the two concepts into one single metric.

* F1 = 2 \* (Precision \* Recall) / (Precision + Recall)
* F1 = 2 \* (0.41 \* 0.50) / (0.41 + 0.50) = 45

**Interpretation:**

* A precision of 0.41 indicates that when the model predicted a failure, it was correct only 41% (i.e., worse than random guessing) of times This indicates the model has had a reasonably low rate of false negatives, creating only few maintenance alerts that are missing from the equipment.
* A Recall of 0.50 shows that the model detected half of all failed one’s realities That 50% detection rate means that while the model successfully discerned some failure events, it missed an additional 50% of them (called false negatives).
* The F1-score of 0.45 suggests the balance between precision and recall is musical, yet unremarkable (True) This score means that the model cannot avoid too many false positives and is not catching all failure events.

## 8.3 Analysis of False Positives and False Negatives

**False Positives:**

* The model produced a significant number of **false positives** (10 cases), which indicates that the predictive maintenance system flagged normal operations as failures.
* In a real-world manufacturing environment, false positives would result in unnecessary maintenance actions, leading to **increased operational costs** and possible **production slowdowns**.

**False Negatives:**

* The model also produced **7 false negatives**, where actual failures were not detected. This is critical because **undetected failures** can lead to unexpected equipment breakdowns and **unplanned downtime**.
* Reducing false negatives is essential for improving the reliability of the predictive maintenance system.

**Recommendations to Address False Positives/Negatives:**

* **Feature Engineering**: Improve the input features (vibration, temperature, pressure) by applying more advanced feature engineering techniques, such as extracting statistical measures (e.g., standard deviation, skewness) or applying time-series transformations.
* **Model Tuning**: Explore hyperparameter tuning for the decision tree model or try alternative models such as **Random Forest** or **Support Vector Machines (SVM)** to improve classification performance.
* **Ensemble Methods**: Use **ensemble learning techniques** (e.g., bagging or boosting) to combine multiple models and reduce the likelihood of false positives and false negatives.

## 8.4. Improve Uptime & plan maintenance periods

In practice, predictive maintenance aims to minimize unexpected downtime by predicting when equipment will have repair requirements (failures). If performance of the model:

* + 1. Expecting numerous false positives would trigger unnecessary as well manual maintenance action, causing a rift in the production schedules and hence an increase in operational costs

1. But despite that, the system had a recall of 0.50 and thus it was able to find some, but not all actual failures (True Positives) so there is still unplanned downtime which could be avoided with preventative maintenance / monitoring solution.

As it stands, the system would probably greatly benefit from additional optimization to lower false positives and raise identification of actual failures so as for even more effective maintenance scheduling.

## 8.5. System Limitations

The results of the evaluation confirmed that while it represents a promising direction, realistic implementation constraints in predictive maintenance are due to several reasons**.**

1. **Data Quality:** High Precision for the virtual model is achieved by using high-quality and assorted data appearances. For example, the model would generalize poorly to real equipment if global databases data is not representative of real-world conditions.
2. **Model Complexity:** The old decision tree model we are using in this implementation may not capture the complexity of real-world failure patterns. However, if you use Random Forests or Neural Networks for this same problem of decreasing churn rate then maybe they provide better performance.
3. **Simulated Environment:** The data used to examine the system was simulated in the non-real-world, which means that its real potential in working conditions remain independent. This could be anything from noise in the real-time data that is not included in your model to any number of other things.

## 8.6. Future Enhancements

Improvements to the predictive maintenance system that might be made include:

* + 1. Advanced Machine Learning: Incorporate more sophisticated machine learning algorithms like Neural Networks, Gradient Boosting Machines (GBM) or Random Forests to make predictions better.

1. Data Augmentation: Add Additional Fault Scenarios and uncertainities in Environment to Increase the variety of simulated data. This will aid the model in generalizing on real-world conditions.
2. Real Time Deployment: Deploy the system in real-time manufacturing set up to validate the effectiveness of it by running on top sensor data and can potentially provide more reliable results what expected.

The process of evaluation for predictive maintenance through the digital twin system, yield some good results however it leaves space for areas where we can enhance. The model accuracy of 43.33%, combined with the observed precision and recall results, demonstrated that there is promise for deploying digital twins in predictive maintenance approaches; however, additional tuning will have to be performed prior to eliminating false positives or increasing failure event detection. When combined with advanced machine learning models and nested into real-world data, this could make it a robust predictive diagnosis system in smart manufacturing to support condition-based maintenance planning or downtime reduction.

# 9. Conclusion and Future Work

This chapter summarizes the main research results and implications for smart manufacturing predictive maintenance based on findings from the dissertation. It concludes with a discussion of several areas for continued study, refinement or enhancement in future implementations of process chains described.

## 9.1. Summary of Findings

The objective of the dissertation was aimed at establishing and testing a MATLAB based digital twin framework for predictive maintenance in smart manufacturing system. The simulation system of the digital twin simulates equipment behavior and predicts failures using machine learning models trained on data from external sensors.

Important conclusions from the study include:

1. With a precision of 0.41 and a recall of 0.50, the decision tree algorithm-based predictive maintenance model produced an accuracy of 43.33%. Although these findings show the model's potential, more work needs to be done to ensure that it performs well enough for practical use.
2. Although the system missed 50% of failures, indicating the need for additional model optimization, it was able to identify 50% of actual failures, which could help minimize unscheduled downtime.
3. A large number of false positives were found in the confusion matrix analysis, indicating that the model frequently produced maintenance alerts that weren't necessary. Although they were less frequent, false negatives still exist and can lead to failures that go unnoticed.

The project demonstrated that the use of **digital twin technology** combined with real-time data and machine learning can offer valuable insights for **predictive maintenance** in smart manufacturing. However, the results also indicated that more sophisticated models and real-world data testing are needed to fully realize the potential of this approach.

The project showed that digital twin technology is capable of complementing predictive maintenance for smart manufacturing by incorporating dynamic real-time data and machine learning. However, the results also suggested that more advanced models and validation with real-world data are necessary to optimize performance of this approach.

## 9.2 Implications for Industry

This dissertation has a number of implications for the manufacturing sector especially opting companies intending to introduce predictive maintenance solutions:

* Less Downtime – The system can identify problems before they disrupt production, helping streamline and reduce downtime of equipment. The model should be further improved in order to reduce false Positives the positives, and that it should not fail detecting actual failures.
* Cost Optimization: Predictive maintenance systems such as the one created in this project can help save money by scheduling maintenance at optimal times, and performing only necessary preventative maintenance which otherwise could have resulted from equipment malfunction.
* Scalability: The MATLAB-compatible digital twin framework can be scaled to integrate many different types of equipment and processes used in manufacturing. Although not fully developed, this could involve progressively more elaborate failure scenarios and environmental conditions inflicting real-time sensor data into the already complex system.

Even if these potential uses could eventually be realized, the system as-is would require additional testing and optimization before it could ever be put into real-world use. This illuminates the need for future work to further improve model performance and translate it in real manufacturing settings.

## 9.3. Limitations of the Study

The current study has limitations that should be addressed in future research.

1. Simulated Data: The model was trained and tested with data generated by the Digital Twin Model provided in MATLAB Simulink. This offered standardized testing settings, but did not replicate the multi-faceted nature of actual factory conditions. Noise or inconsistencies in sensor data can lead to a difference with the performance of models on real-world applications as well.
2. Model Simplicity: This study uses decision trees which are a relatively simple model and might not be able to capture the full complexity of failure patterns in manufacturing equipment. Note that more powerful machine learning models (e.g., deep-learning-based) such as neural networks, or ensemble methods might perform better.
3. Few Failure Scenarios: The simulation was a bit light on failure scenarios. In real-world scenarios, equipment failure can occur due to numerous factors and the model's ability at predicting them is only as good as how robustly represented these examples were in the training data.

## 9.4. Future Work

This section suggests future work to improve the predictive maintenance system and increase its applicability for real-world manufacturing environments.

**Integration With Real-time data**

The natural path forward would be to get the predictive maintenance system running on real sensor data from live manufacturing hardware. Finally, live sensor data will allow verification and real-time validation of how the system is performing as well as give us an idea about characteristic generalization to actual equipment behavior.

**Advanced Machine Learning Techniques**

In the future, improvements on machine learning models could be made by using more advanced techniques which increase accuracy and stability of the system like:

* + 1. Random Forests or Gradient Boosting Machines (GBMs) would be able to model more complex decision boundaries and often result in higher generalization than a single tree.
    2. Advanced Neural Networks and Deep Learning models are the most suitable for detecting complex non-linear patterns in sensor data, which correlate with potential equipment failures.
    3. Ensemble Learning: Using more than one model at the same time (e.g. a combination of decision trees, SVMs and/or neural networks) may decrease false positive as well ass false negative errors, thus making our predictions much robust

**Information Overload data Augmentation and Feature Engineering**

Higher quality and diversity in training data are more important measure for the improvement of primary failure detection Future work could focus on:

* Performing Data Augmentation: It may be introducing even more failure scenarios or utilizing data generation methods to enhance the dataset.
* Feature Engineering: Converting the sensor data into more insightful features like rolling statistics, moving averages and other frequency domain characteristics could provide a direct fillip to your model.

**Real-Time Dashboard Improvements**

MATLAB has been employed to implement a real-time dashboard with an easy-to-understand interface that is capable of monitoring the health status and issuing predictive maintenance alert concerning equipment. Future work could:

* + 1. Provide interactive dashboard components, e.g., allowing operators to drill-down into past data or dive deep into specific failure modes.
    2. Alert prioritization to display the most critical alerts prominently and limit false positives.
    3. Make the dashboard accessible on mobile or web to provide equipment operators with a way of monitoring it remotely

**Cost-Benefit Analysis**

Another relevant issue for future research is the cost-effectiveness of a predictive maintenance system in real industrial settings. This may consist of using calculations that measure reduced downtime, an increase in time between maintenance schedules or a decrease in equipment breaking down over and over again as a result. Insights into this analysis would be invaluable for decision-makers looking to launch predictive maintenance solutions.

## 9.5. Final Thoughts

The research conducted in this dissertation highlights the benefits of using digital twin technology with predictive maintenance models to achieve more reliable equipment and reduce unexpected downtime in smart manufacturing. Although the results also suggest that more work will be necessary to improve the system's performance, they represent a good basis for future research and development.

The use of real-time data-driven maintenance tactics for the factory sector will be an indispensable solution to improving industrial operations and competition in a world where manufacturing systems keep getting more intricate. With newer machine learning models rolled into a predictive maintenance system that utilizes real-time data and adapts to individual machines, manufacturers may finally have the scalable digital twin framework they sought for this kind of use-case.

# 10. Contributions of team members

1. Himanth Sai Jalagam - Y00860828
   1. Project Coordination and Planning
   2. Simulink Model Development
   3. Data Integration and Preprocessing
   4. Documentation
2. Ram Charan Araja - Y00857875
   1. Literature Review and Background Research
   2. Predictive Maintenance Algorithm Development
   3. Performance Evaluation and Analysis
   4. Report Writing
3. Durga Reddy Polimera - Y00856583
   1. Real-Time Dashboard Development
   2. Data Visualization and User Interface
   3. Testing and Debugging
   4. Final Documentation and Presentation

# References

1. Grieves, M. (2014). Digital twin: Manufacturing excellence through virtual factory replication. Retrieved from <https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication>
2. Glaessgen, E., & Stargel, D. (2012). The digital twin paradigm for future NASA and US Air Force vehicles. 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. Retrieved from <https://ntrs.nasa.gov/citations/20120008178>
3. Tao, F., Zhang, H., Liu, A., & Nee, A. Y. (2019). Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics, 15(4), 2405-2415. <https://doi.org/10.1109/TII.2018.2873186>
4. Negri, E., Fumagalli, L., & Macchi, M. (2017). A review of the roles of digital twin in CPS-based production systems. Procedia Manufacturing, 11, 939-948. <https://doi.org/10.1016/j.promfg.2017.07.198>
5. Qi, Q., & Tao, F. (2018). Digital twin and big data towards smart manufacturing and Industry 4.0: 360-degree comparison. IEEE Access, 6, 3585-3593. <https://doi.org/10.1109/ACCESS.2018.2793265>
6. Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing Letters, 3(1), 18-23. <https://doi.org/10.1016/j.mfglet.2014.12.001>
7. Mobley, R. K. (2002). An introduction to predictive maintenance. Butterworth-Heinemann. Retrieved from <https://www.elsevier.com/books/an-introduction-to-predictive-maintenance/mobley/978-0-7506-7531-4>
8. Shao, G., & Helu, M. (2020). Framework for a digital twin in manufacturing: Definition, design, and implementation. Journal of Manufacturing Systems, 58, 179-192. <https://doi.org/10.1016/j.jmsy.2020.06.015>
9. Bokrantz, J., Skoogh, A., Berlin, J., Stahre, J., & Kristoffersen, E. (2020). Data-driven decision support for maintenance: Lessons learned from manufacturing industry. Journal of Manufacturing Science and Engineering, 142(9), 1-9. <https://doi.org/10.1115/1.4047298>
10. Khajavi, S. H., Motlagh, N. H., Jaribion, A., Werner, L. C., & Holmström, J. (2019). Digital twin: Vision, benefits, boundaries, and creation for buildings. IEEE Access, 7, 147406-147419. [https://doi.org/10.1109/ACCESS.2019.2946515](%20https:/doi.org/10.1109/ACCESS.2019.2946515)
11. Rosen, R., Wichert, G., Lo, G., & Bettenhausen, K. D. (2015). About the importance of autonomy and digital twins for the future of manufacturing. IFAC-PapersOnLine, 48(3), 567-572. <https://doi.org/10.1016/j.ifacol.2015.06.141>
12. Wang, L., Törngren, M., & Onori, M. (2015). Current status and advancement of cyber-physical systems in manufacturing. Journal of Manufacturing Systems, 37, 517-527. <https://doi.org/10.1016/j.jmsy.2015.04.008>
13. Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges, and open research. IEEE Access, 8, 108952-108971. <https://doi.org/10.1109/ACCESS.2020.2998358>
14. Uhlemann, T. H. J., Lehmann, C., & Steinhilper, R. (2017). The digital twin: Realizing the cyber-physical production system for industry 4.0. Procedia CIRP, 61, 335-340. <https://doi.org/10.1016/j.procir.2016.11.152>
15. Lindström, V., Malmgren, T., & Jonsson, M. (2020). Challenges of digital twins for predictive maintenance of production systems—A comprehensive literature review. Journal of Manufacturing Technology Management, 31(6), 1199-1220. <https://doi.org/10.1108/JMTM-12-2018-0446>

Appendix:

* 1. GitHub Repository link: <https://github.com/himanth2028/project-Digital-Twin>

PROJECT GUIDE APPROVAL SCREENSHOT:

A screenshot of a computer

Description automatically generated