An Applicable Approach to Digital Twins for Precision Measurement and Quality Assurance

Chas Hamel

DSA5900 – Spring 2024 (4 Credits)

Sponsor: Dr. Shiva Raman, Department of Industrial & Systems Engineering

Table of Contents

Section 1: Motivation

If asked about an experience where I made a mistake at work, I will always talk about a “conical hole plug” that failed in production creating a projectile in a manufacturing environment. The conical plug **(Figure 1 - Item 1)** was designed to fit inside of a conical hole **(Figure 1 - Item 2)** that, when compressed by a 1-1/2” – 6 Grade 8 Bolt **(Figure 1 - Item 3)**, created a seal that held elastomer being extruded into a steel tube at a maximum of 1400 bar (22,000psi) from leaking through the hole. The pressure is depicted by red arrows in **Figure 1**. During assembly of this plug assembly, the bolt, which is reused each extrusion process, is coated with a lubricating copper coating to aid in the bolt being properly torqued while compressing the plug.

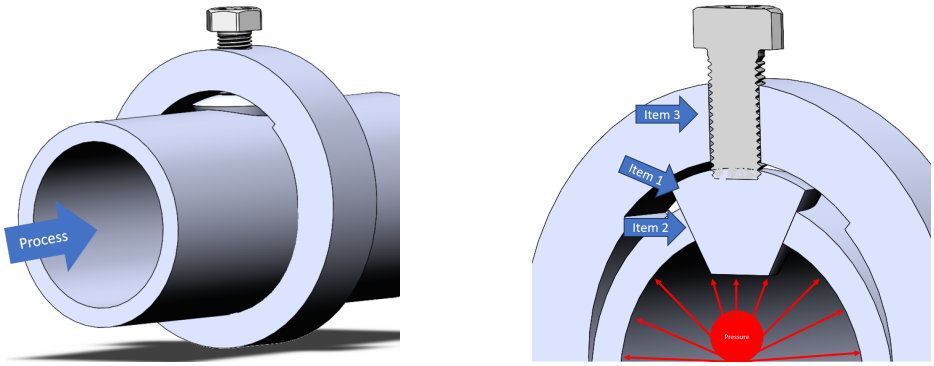

Figure 1: Conical Hole Plug Assembly

During the elastomer extrusion process, the inside of the steel tube and resultant Conical Hole Plug sees extreme outward pressure, applying an upward force onto the threads of the 1-1/2” – 6 Grade 8 Bolt. After repeated assembly, processing cycles, and disassembly, the threads of the bolt begin to fatigue and deform like the region in red of the bolt seen in **Figure 2**. This deformation ultimately leads to thread failure, allowing pressure loss from inside of the steel tube thrusting the 1-1/2” – 6 Grade 8 Bolt uncontrollably into areas populated by manufacturing operators.

Figure 2: Deformed vs Acceptable Threads

In this scenario, a robust thread inspection process paired with continuous evaluation of extrusion process data, the Manufacturing Engineering Team could have predicted bolt failure and performed preventative maintenance prior to putting manufacturing personnel in harm’s way.

Section 2: Introduction

In manufacturing, it is no secret that the industry is in the middle of the Fourth Industrial Revolution. This revolution, coined “Industry 4.0”, is the transformation of archaic manufacturing processes and process controls into smart, connected manufacturing systems that are developed, optimized, and controlled using digital technologies such as the Internet of Things (IoT), Big Data, Machine Learning, and other technologies that can increase product quality, product reliability, and production capacity while lowering operating costs to the production facility.

One aspect in the manufacturing process that is often scrutinized is the quality control of products produced as, from a customer perspective, these tasks do not add any value to the overall product being produced. The constant push and pull within a manufacturing facility to confirm that product is produced to specification while limiting the amount of time machines and operators “touch” the part is a top priority for the manufacturing engineering teams.

A transformative idea to reduce inspection time while aiding in the increase in product quality and its predicted reliability is the use of a “Digital Twin” for inspection equipment such as a Coordinate Measurement Machine (CMM), Vision Inspection Systems, and other software driven equipment.

A Digital Twin is the use of a digital model that simulates a Physical Twin, which represents a physical process, environment, assembly, or component. Relative to digital inspection equipment, the Digital Twin is used to generate and confirm the process the digital inspection equipment shall follow prior to implementation, eliminating the time required for a Quality Inspector to manually program these processes while ensuring the part being inspected is properly reviewed.

Additional to the inspection processes, a Digital Twin of a part or assembly manufactured and inspected through the entire product’s lifecycle can allow Machine Learning and Predictive Models to guide engineers in making decisions on preventative maintenance while in operations prior to product failure.

To make a shift in the industrial mindset, the theoretical and academic concepts of how data, systems, and humans interact need to be applied and operationalized in an efficient manner. The Applicable Approach to Digital Twins for Precision Measurement and Quality Assurance study provides a real-world overview to the implementation of this methodology in a manufacturing facility and provide a model that will allow operations and reliability engineers to make data driven decisions on how to maintain products operational in the field.

Section 3: Objectives

This project aims to explore the use and benefit of Digital Twins within manufacturing and to provide a practical implementation of a Digital Twin to represent the CyberOptics CyberGage 360 Vision Inspection System[1] (CyberGage 360) and a 3D Printed 1”- 4 ACME Bolt (Bolt). After the Digital Twin for both items are implemented, data ingested by the Digital Twin of the Bolt shall reflect the environment and actions performed on it and can be used to predict future performance and failures.

To accomplish the objective, the following specific objectives have been established:

1. **Implement Digital Twins to represent the CyberGage 360 and 1”- 4 ACME Bolt.**
   For the first objective, a Digital Twin shall be generated using SolidWorks and Microsoft’s Azure Digital Twins Environment to represent the CyberGage 360 and Bolt respectively. Model Schema for each Digital Twin shall be established based on mechanical 3D models of the Equipment and Critical to Function Dimensional Measurements of the Bolt.

2. **Establish a robust workflow and data stream to utilize the Digital Twin of the CyberGage 360 to confirm the inspection process of the Bolt and to ingest fatigue testing data and dimensional inspection results of the Bolt into its Digital Twin.**
   For the second objective, the Digital Twin of the CyberGage 360 in SolidWorks will be utilized to find an efficient process to perform and confirm the inspection of Critical Dimensions for the Bolt. Additionally, data streams from the Physical Twin of the Laboratory based CyberGage 360 and Fatigue Testing Equipment shall be established to feed testing data and resultant dimensional inspection data to the Azure based Digital Twin of the Bolt.

3. **Using data ingested by the Bolt Digital Twin, implement a Lifecycle Predictive Model to predict failures of the Physical Twin to advise when preventative maintenance shall be performed.**
   For the third objective, repetitive cycles of fatigue testing and dimensional inspection shall occur using the Physical Twins. The data from these cycles shall feed the established Digital Twins resulting in a database of mechanical test data and dimensional data. This data shall be used to build Machine Learning Models to aid in predicting failures in the Physical Twins that the Digital Twin represents.

Section 4: Data Ingestion, Exploration, and Preparation

CyberGage 360 Digital Twin

Prior to data ingestion into the Digital Twin of the Bolt via data streams from the Laboratory based Test & Inspection Equipment, it must first be established that the part being evaluated is capable of a full dimensional inspection using the CyberGage360. The CyberGage 360 uses image data from 6 total image sensors, two of which are, according to CyberOptics, "dual camera optical blue light scanning sensors mounted above and below the subject part sitting on an optically flat, clear glass plate calibrated for scanning. The glass plate allows simultaneous data capture from both sensors and eliminates the need to flip-over the part necessary for all other scanning and other conventional measuring systems."[2]

Given that the system uses cameras from different angles to simultaneously scan a component that must sit at the center of the 157,909 mm^2 area rotating glass plate within a 200mm diameter x 100mm high cylinder (31,415.93 mm^2 Area), there is a ~80% chance that a portion of the component will sit outside of the inspection area when placed on the glass inspection plate**. Figure 3** provides an overview of the Digital Twin and Physical Twin of the CyberGage360 where, on the Digital Twin, the green assemblies represent the six image sensors, the blue plate represents the optically clear rotating inspection plate, and the red cylinder represents the 200mm diameter x 100mm cylindrical inspection area.

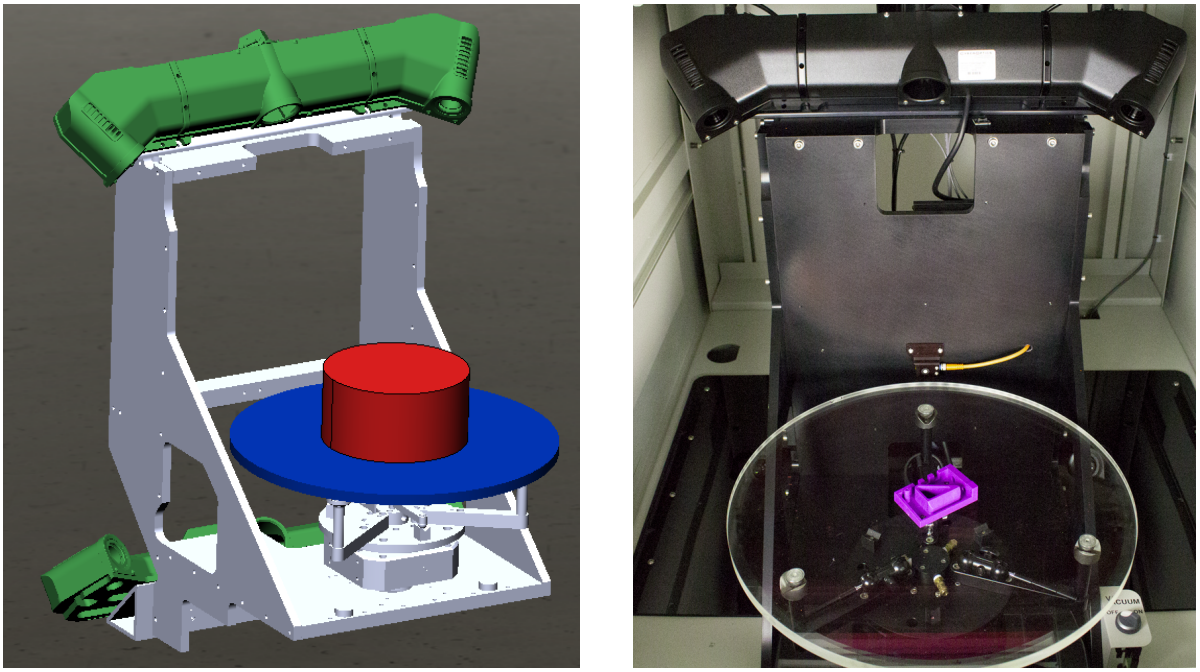

Figure 3: Left – CyberGage360 Digital Twin, Right – CyberGage360 Physical Twin[3]

Due to this potential error and to ensure an accurate inspection of all critical features of the Bolt Physical Twin, the CyberGage 360 Digital Twin allows Manufacturing and Quality Engineers to develop an inspection process that is part specific and is validated to be accurate prior to releasing to production.

For the Bolt selected, a Nylon-12 3D Printed 1" – 4 ACME Bolt, an Interference Detection process is performed between the Bolt and the cylindrical inspection area to confirm that the part fits within this envelope when sitting in the center of the rotating inspection plate, identified by the red region of the bolt in **Figure 4**. Alternatively, if the bolt was placed 225mm to the left, the Interference Detection process will show that the Bolt is not fully enveloped by the inspection area, identified by the silver portion of the bolt in **Figure 5**, and should not be placed in this region in production.

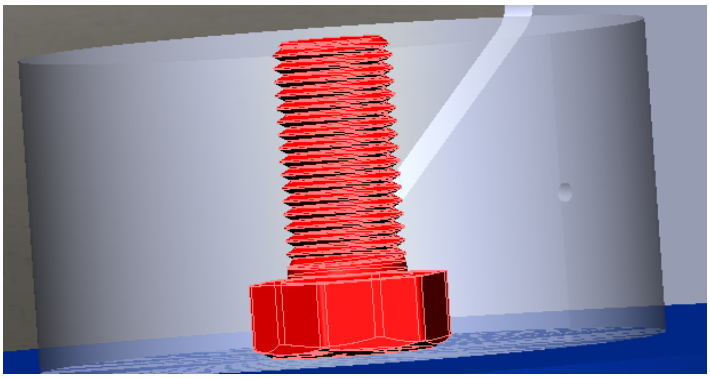

Figure 4: Bolt Inside of Inspection Region (red)

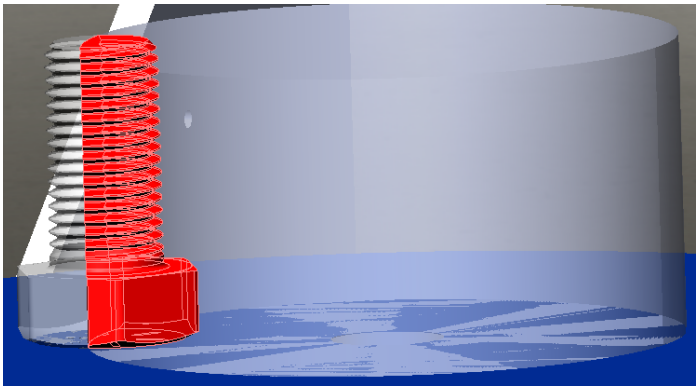

Figure 5: Bolt Outside of Inspection Region (silver)

Now that the CyberGage 360 Digital Twin has confirmed the Bolt will fit into the inspection region of the CyberGage360, it must be confirmed that all critical features of the Bolt can be seen by the six image sensors in the orientation confirmed the bolt will be placed into the inspection region.

The process to confirm all critical features of the Bolt can be inspected uses a "light cone" from a single image sensor is seen in **Figure 6**. Using the interference between the light cone and the part, engineers

can determine if there are hidden shadows on the part to which the image sensors cannot see, such as through overhands or internal features.

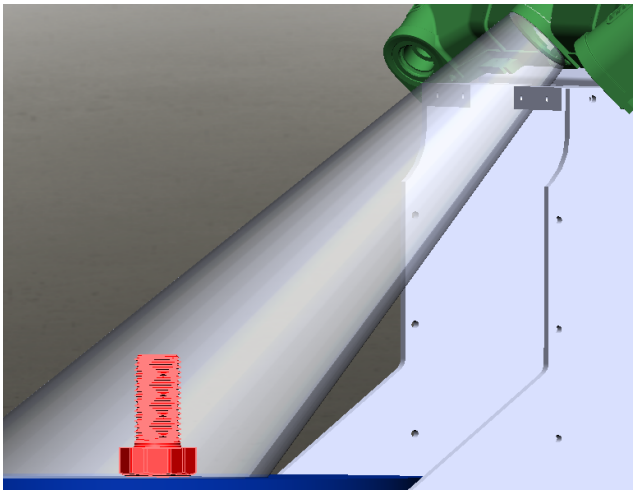

Figure 6: Interference Detection between Bolt and Image Sensor

Bolt Physical Twin

For the Physical Twin of the 1”-4 ACME Bolt, a 3D Printed component is designed in 3D CAD Solidworks as shown in **Figure 7**. This 3D model is fully defined using Model Based Definitions where the critical features are automatically identified and labeled by the software.

Using powdered Nylon-12 material, the bolts are 3D printed using Selective Laser Sintering (SLS) manufacturing methods. The Nylon-12 material was selected for its high tensile strength (7,252 psi) and an overall 4% elongation at break, giving it a designation as being a brittle material compared to ductile. **Figure 8** shows the final manufactured component used for the remaining portions of this project.

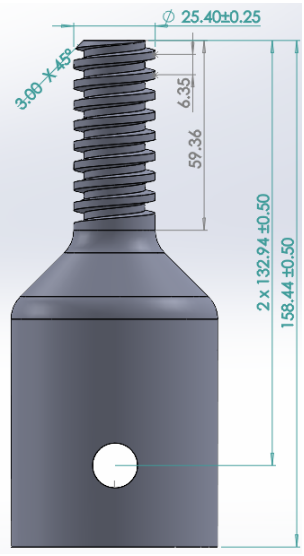

Figure 7: Fully Defined 3D CAD Model

Figure 8: 3D Printed Bolt Physical Twin

Using the fully defined 3D CAD model from Solidworks, the baseline dimensional data is fed into Geomagic Control X software, a highly capable dimensional comparison tool used for comparing 3D Laser Scanned models to a theoretical perfect model to visualize deviations and discrepancies in the part being inspected. Using the model-based definitions from the 3D CAD Solidworks model, the nominal dimensional data for each critical feature is loaded into the software to be used when comparing 3D Scan data from the CyberGage360. **Figure 9** below shows a cross section of the 1”-4 ACME Threads with each critical feature defined with a nominal dimensional measurement and tolerance. After laser scanning each component, a comparison is performed with heatmap, and actual dimensional data visualized as depicted in **Figure 10**.

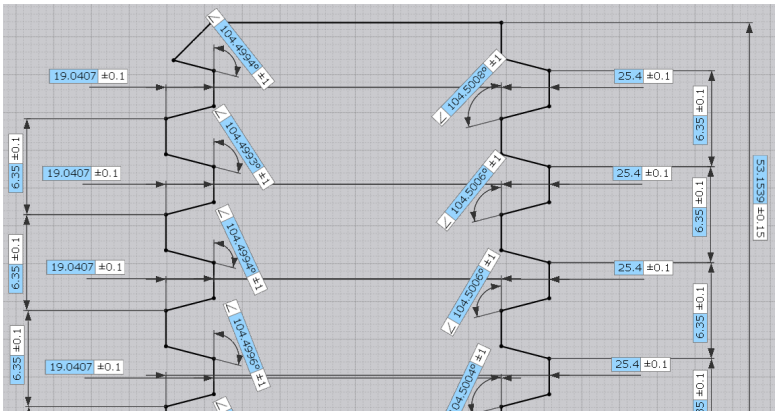

Figure 9: Theoretical Nominal Dimensions in Geomagic Control X Software

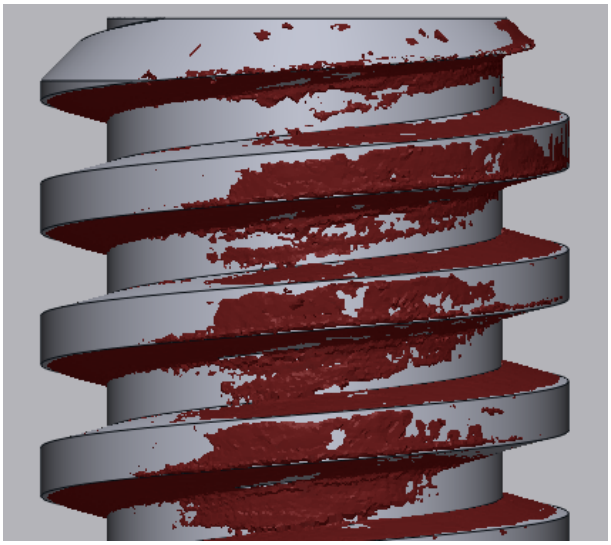

Figure 10: Example Comparison of 3D CAD and 3D Laser Scan Data

Bolt Digital Twin

As observed in the previous section, a 3D Model representing the Bolt is used to confirm all inspection processes within the CyberGage 360 Digital Twin. As future predictive models in this project will use data representing fatigue testing performed on the Bolt and dimensional data output from the Physical Twin of the CyberGage 360, a Digital Twin model shall be developed using the Digital Twin environment in Azure.

Within Azure, a Digital Twin model can be input by generating a JSON file containing the Digital Twin Definition Language (DTDL)[4]. For each Digital Twin representing a Physical Twin, a model shall be input into Azure with the example schema demonstrated below in **Table 1**.

Table 1: Bolt Digital Twin Schema Example

name	@type	schema
Overall_Length	Property	Float
Major_Diameter_1	Property	Float
Angle_Left_1	Property	Float
Max_Load	Property	Float
Max_Position	Property	Float
Fracture	Property	Boolean

A portion of the Azure representation can be seen below in **Image 11**.

Figure 11: Digital Twin Schema in Azure

For this project, 10 Bolt Digital Twins representing 10 Bolt Physical Twins are generated using the schema provided above and initialized with preliminary inspection data from the Physical Twin CyberGage 360. An example of the loaded properties of Bolt #1 are below in **Figure 12.**

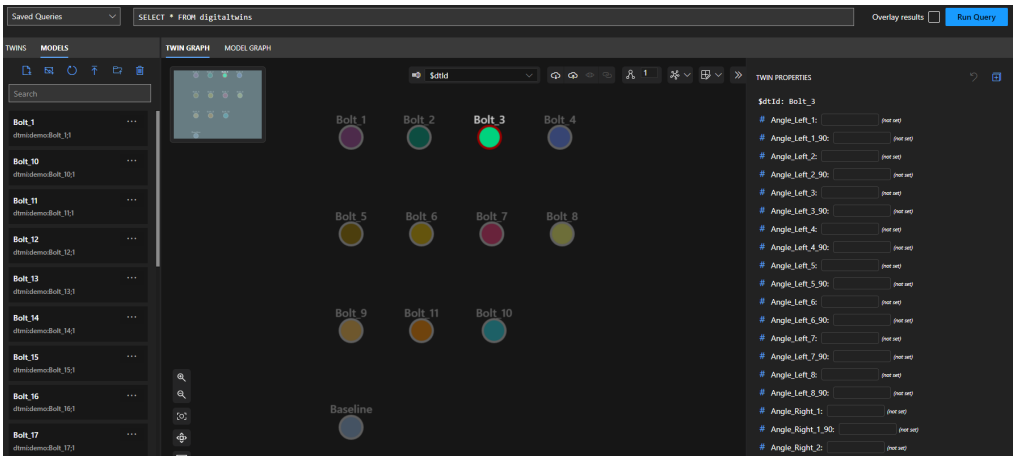

Figure 12: Initialized Digital Twins of Bolt, with Bolt #3 Data Visualized

Data Stream

With the Digital Twin of the Bolt generated, tensile testing of the 10 Bolt Physical Twins shall be performed using the test plan defined in **Table 2** for a minimum of 10 cycles or until Bolt failure. The set up for each bolt on the tensile testing machine is seen in **Figure 13**. For each Bolt and each step of the test plan, real-time logging of data from the tensile testing and dimensional inspection shall occur and ingested by the relevant Digital Twin in Azure.

In the following table **(Table 2),** the test plan depicts the tensile elongation, in inches, each bolt will experience for each tensile test**.** After each test, a dimensional scan using the CyberGage360 is performed.

Table 2: Data Collection Test Plan

	Bolt 1	Bolt 2	Bolt 3	Bolt 4	Bolt 5	Bolt 6	Bolt 7	Bolt 8	Bolt 9	Bolt 10
Test 1	0.055	0.055	0.052	0.051	0.047	0.05	0.05	0.045	0.055	0.05
Test 2	Failure	0.04	0.049	0.046	0.049	0.051	0.047	0.045	0.047	0.052
Test 3	Failure	Failure	0.044	0.052	0.047	0.041	0.051	0.057	0.04	0.049
Test 4	Failure	Failure	0.045	0.049	0.053	0.055	0.058	0.041	0.044	0.047
Test 5	Failure	Failure	0.043	0.055	0.059	0.045	0.054	0.06	0.053	0.054
Test 6	Failure	Failure	0.063	0.062	0.061	0.06	0.064	0.064	0.064	0.06
Test 7	Failure	Failure	0.064	0.065	0.062	0.063	0.067	0.068	0.067	0.062
Test 8	Failure	Failure	0.064	0.07	0.063	0.062	0.069	0.065	0.064	0.063
Test 9	Failure	Failure	0.068	0.064	0.066	0.07	0.068	0.07	0.069	0.072
Test 10	Failure	Failure	0.074	0.76	0.076	0.74	0.078	0.079	0.075	0.078
Test 11	Failure	Failure	0.08	Failure	0.081	0.081	0.083	0.084	0.083	0.082

After each test cycle, the model will consist of tensile test data and dimensional data for each bolt and test combination. Each Bolt Physical Twin will have been tested to Failure as to ensure an accurate information is fed into the Machine Learning models with tensile loads and elongation experienced during testing and the resultant dimensional data after the testing occurred.

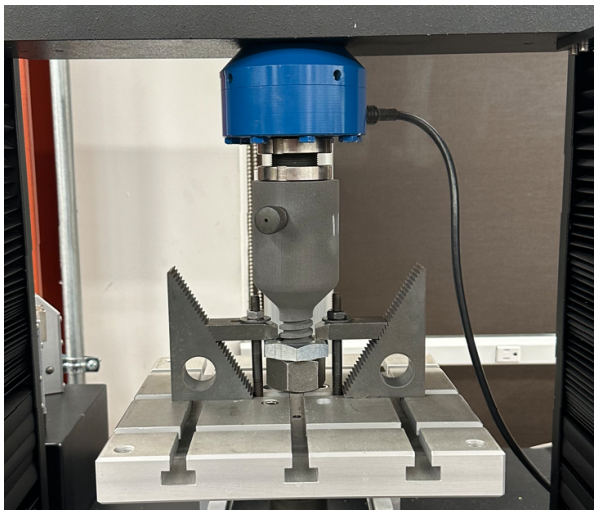

Figure 13: Physical Twin Bolt Set Up for Tensile Testing

Section 5: Methodology

Example – Imaginary Company's Digital Twin Implementation

With the intent of this project to be an applicable proof of concept of the relationship between a Digital and Physical Twin using a simple component such as a Bolt, we should first step back and understand the impact having a Digital Twin can have on understanding a much larger Physical System.

Assume Imaginary Company has a chemical processing plant with various types of assets, such as pumps, compressors, and turbines. Each asset is critical to the production output of Imaginary Company, and any unplanned downtime leading to a significant financial loss for the company. To better understand each asset and the environmental conditions it sees in real-time, Imaginary Company implements a Digital Twin of each asset in Microsoft's Azure Digital Twin environment. Physical sensors located on each asset are connected to Imaginary Company's network and feeds data in real-time, with up to 10 datapoints / second, of parameters that the Process Engineers find critical to the overall functionality of the asset, such as;

- Rotational Speed (revolutions per minute -RPM),
- Axial Vibration of the Rotor (Hertz – Hz),
- Process Flow Rate (liters per minute – l/m),
- Process temperature (Celsius - °C)
- Axial Bearing Gap, via periodic inspection (millimeters – mm)

Using this data, the Process Engineers analyze a Supervised Machine Learning Random Forest model that was developed using data from historical assets that are currently in service and from those that have previously failed, resulting in production shutdowns.

As the Digital Twins are continuously fed real time data, Process Engineers can understand predicted time to failure (hours) of each asset, based on its asset type and processing parameters, as to allow Imaginary Company's maintenance teams to be able to prepare and plan for each assets maintenance without being impacted by sudden equipment failure.

Additionally, the Process Engineers can feed theoretical data, such as increasing the Process Flow Rate by 15% and Rotational Speed by 10% to increase production output over 90 days, the Process Engineers can understand how this would impact the total lifetime of the equipment prior to making the decision.

It is the final piece of this example, where the Random Forest model is developed and utilized to predict time to failure of components that the remaining portions of this project will explore.

Data Preparation

With all data loaded into the Azure Digital Twin database from two distinct sources, feature engineering, missing value reconciliation, outlier handling, as well as bootstrapping occurred on the complete dataset to ensure optimization for modeling.

Feature Engineering

For each combination of bolt and test number, the database consists of 97 unique dimensional measurements and thousands of rows of real time Force (lbs) and Elongation (inch) data from tensile testing. As characteristic regression models perform best with unique rows of data which summarize the entry, feature engineering occurred to calculate critical features, such as Max Force and Max Elongation for each test combination. Furthermore, for mechanical components, tensile testing reveals characteristics of the test component and its mechanical properties. With the raw data from the tensile testing machine being Force (lbs) and Elongation (inch), the Stress and Strain the material experienced can be calculated to allow further Material Science based conclusions to be drawn. Using Equation 1 and Equation 2 respectively, the average and maximum stress and strain are calculated and appended to the dataset.

$$Eq.1 \quad \sigma = \frac{F}{A} \; where;$$

F is the force applied,
A is the cross-sectional area of the component

$$Eq.2 \quad \epsilon = \frac{\Delta L}{L_0} \; where;$$

ΔL is the change in length,
L_0 is the initial length

Lastly, for Feature Engineering, a binary feature depicting if the bolt failed or not within the test cycle is generated and is used as the Target Variable later in the project.

Missing Values

Although initially believed to be unlikely due to the nature of the data being generated from data loggers on laboratory process machines, an evaluation of the database occurred to ensure all missing values are rectified. As shown in **Figure 14**, the dataset consists of four columns with missing values, with 1 or 2 missing values each, which with equates to 5.8% of the 86 rows within the dataset having missing values.

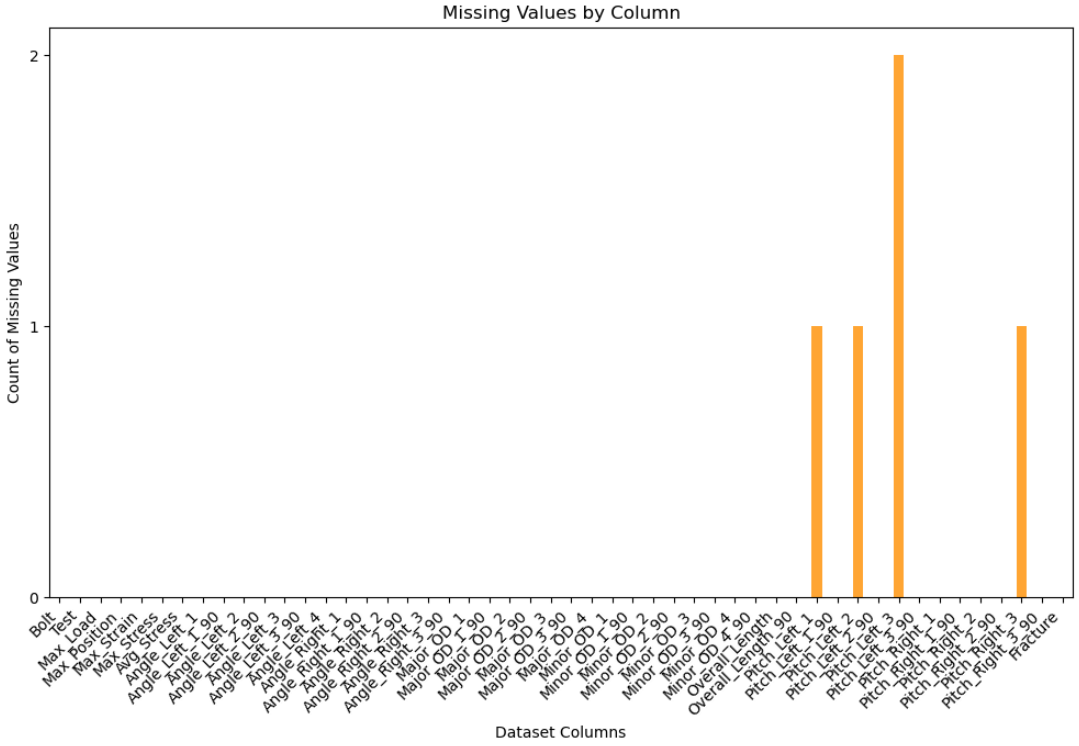

Figure 14: Missing Values within the Dataset

The entirety of the missing values is within the dimensional inspection portion of the overall dataset thus conclusions can be drawn to provide accurate imputation of data. As each measurement, such as Overall Length, is taken on two different locations to ensure accuracy, the missing values are handled by replacing the missing value with the similar measurement taken 90 degrees from the missing value for the same bolt and test combination. For example, one bolt on one test had a missing value within the "Pitch_Left_1" field and does not have a missing value within the "Pitch_Left_1_90" field. As the "_90" dimension is similar to the field with missing value, the dimension is imputed into the missing field.

Outliers

With the dataset complete with no missing values, an evaluation of outliers was performed using the statistical z-score where an item is considered an outlier if the z score absolute value is greater than 3.0,

excluding the Target Variable column, “Fracture”. As depicted in **Figure 15**, the box plots for each Feature column give reference to the quantity of outliers per feature, where the quantity of values with an absolute z score greater than 3.0 are found.

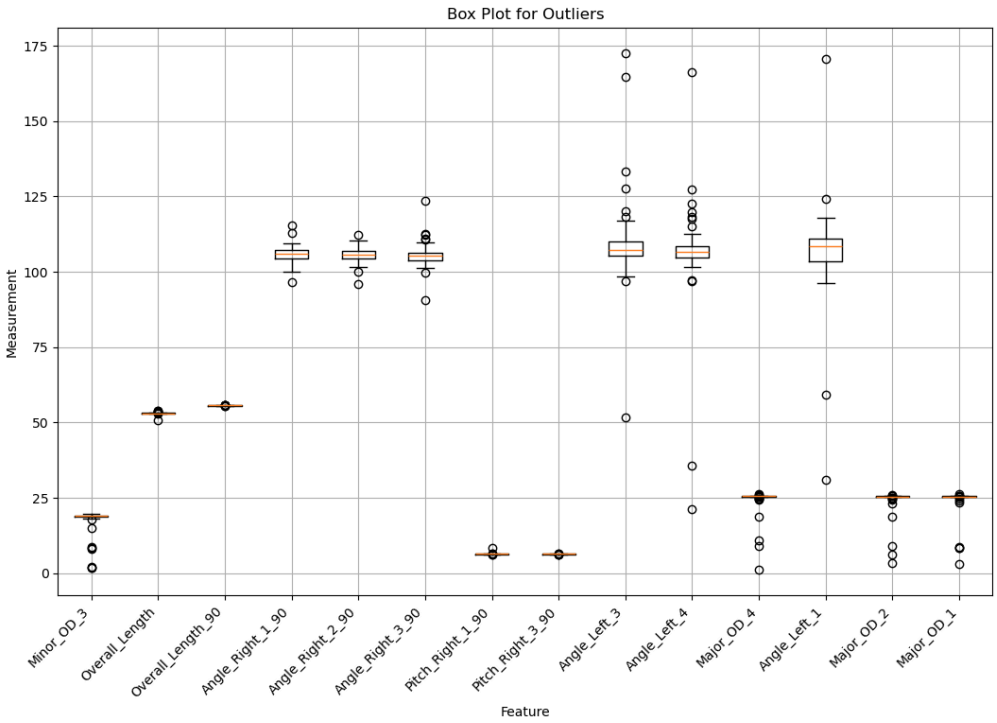

Figure 15: Outlier Values per Feature

To reduce the impact of each outlier without generating a missing value, data imputation occurred by taking the average value of that feature for all tests performed on that specific bolt prior to the outlier occurring. By using the average of previous tests, and not of the entire dataset, the statistical integrity of the dataset specific to each bolt is held.

Bootstrapping

As this project focuses on generating real-world data on a Physical Twin, the overall sample size of and total data frame length is relatively small compared to datasets generally used in academia. With an initial data frame length of 86 rows, bootstrapping, or data augmentation, was a necessary action to perform to ensure that the overall dataset ingested into the machine learning model was large enough for proper training to occur while holding the statistical integrity of the data intact. By ensuring data replacement, the number of bolts in the data frame is increased to n = 100, with the number of tensile tests performed per bolt set to a maximum of i = 11. This action significantly increased the overall length of the data frame for modelling to 1100 rows.

Data Exploration

Correlation Matrix

With the dataset cleaned and prepped for modeling, the first evaluation performed on the entire dataset is a correlation matrix to visualize the relationships between each feature. **Figure 16** below shows the output heat map correlation matrix for the entire dataset.

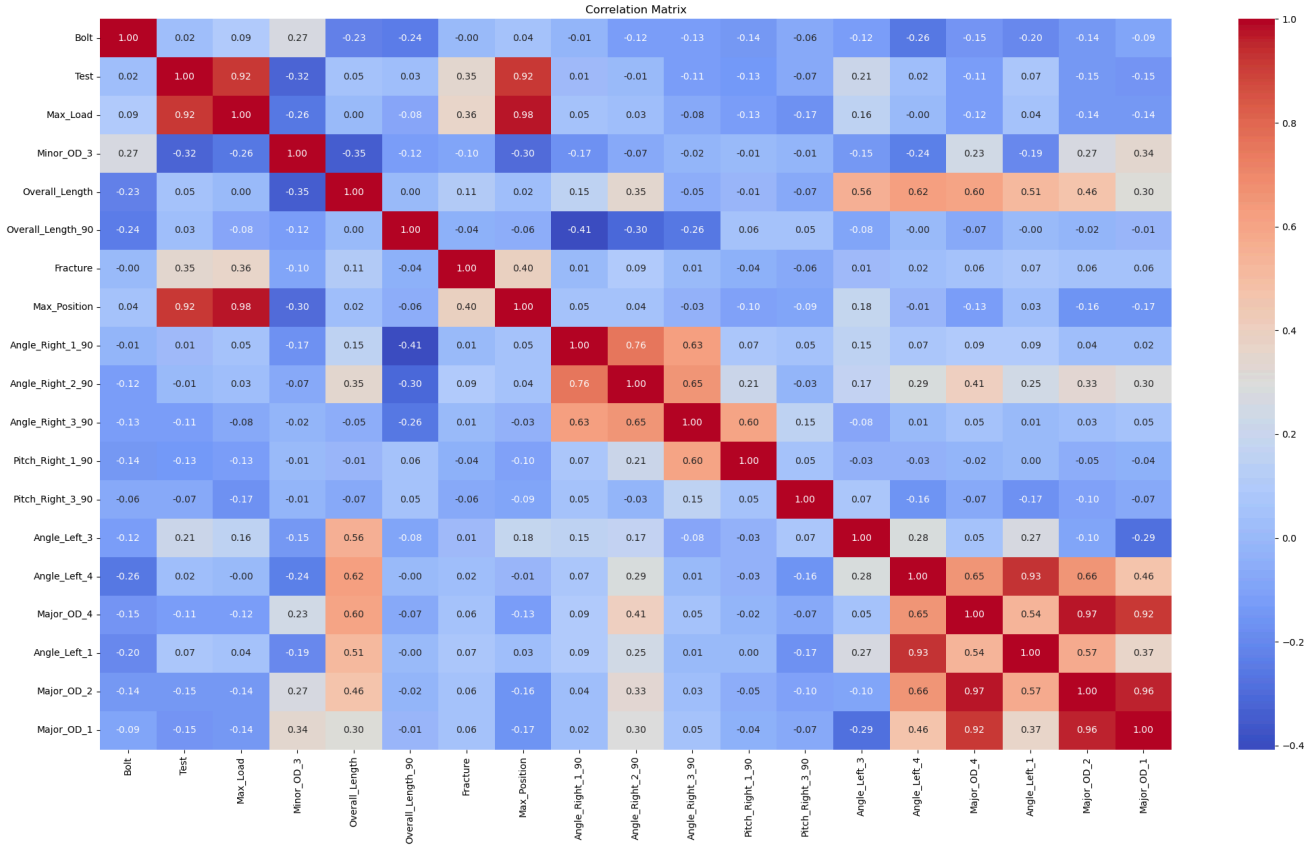

Figure 16: Heat Map Correlation Matrix

From this matrix, we can visualize obvious relationships such as the Max Load applied to the bolt in relation to the Max Position achieved during test, as these are directly related. Relationships that are not as obvious are a Fracture, or failure in testing, is loosely correlated to the Max Load and Max Position achieved by the bolt in tensile testing. Additionally, of significant interest is the correlation between Overall Length and Fracture. Although extremely low, at 0.11, this relationship is significantly higher in the positive direction than any other dimensional measurement to the overall prediction of if the bolt failed or not.

Descriptive Statistics

Using descriptive statistics to understand the data structure further, the standard deviation of each feature is evaluated. As shown in **Table 3**, the Top 5-Dimensional Measurements with the highest standard deviation are observed.

Table 3: Top 5-Dimensional Measurements with Highest Standard Deviation

Feature	Angle_Right_1	Angle_Right_2	Angle_Right_3	Angle_Left_4	Angle_Left_2
Standard Deviation (degrees)	20.19	19.73	18.76	14.24	13.24

As illustrated in **Figure 17**, these five measurements with the highest standard deviation are all thread angles where the highest downward load is applied by the tensile testing machine. Although a quick assumption could be made that there is a large standard deviation to these thread angles as deformation is occurring during testing, which would benefit the overall conclusion of this project, it is unclear if this is factual or if the measurement system is inadequate in measuring thread angles as the top 11 features with an elevated standard deviation are all angle measurements. As a standard ACME thread was used for test specimens, the thread angle is highly controlled. Thus, it is likely that this deviation is caused by inconsistencies in the 3D Printing Manufacturing Process of the bolts or within the measurement system compared to being a result of tensile testing.

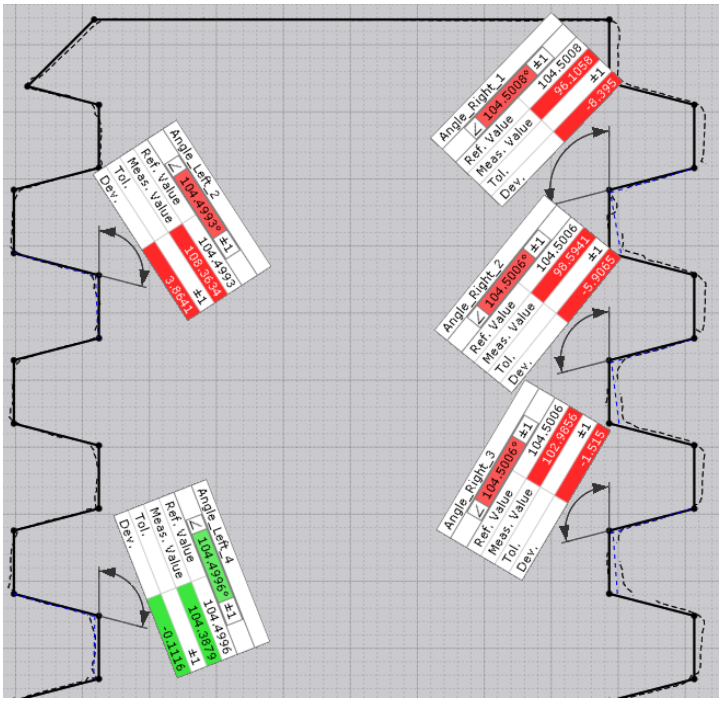

Figure 17: Bolt Cross Section with Dimensions of High Standard Deviation

K-Means Clusters

To further understand the relationship in the bootstrapped dataset, a K-Means Clustering Analysis was performed using Principal Components of the entire dataset to reduce dimensionality. Observing the elbow curve as an output from the K-Means Clustering Technique, an initial conclusion is drawn that there are six distinct clusters based on an apparent "elbow" being present as shown in **Figure 18**. As the Principal Components within the clusters were visualized, it became obvious that there was not enough distinction between the clusters, as shown in **Figure 19**, thus the number of unique clusters was reduced to three. Using the reduction in clusters, the Principal Component graph in **Figure 20** was generated to show a clear distinction in the clustering of the bootstrapped dataset where 50.3% of the data is in Cluster 0, 48.3% of the data is in Cluster 1, and 1.2% of the data is in Cluster 2.

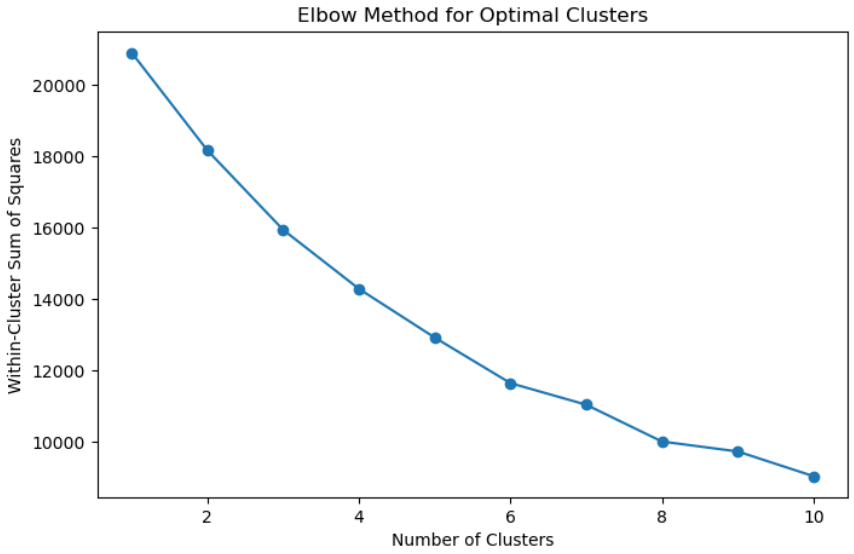

Figure 18: K-Mean Cluster Analysis for Number of Clusters

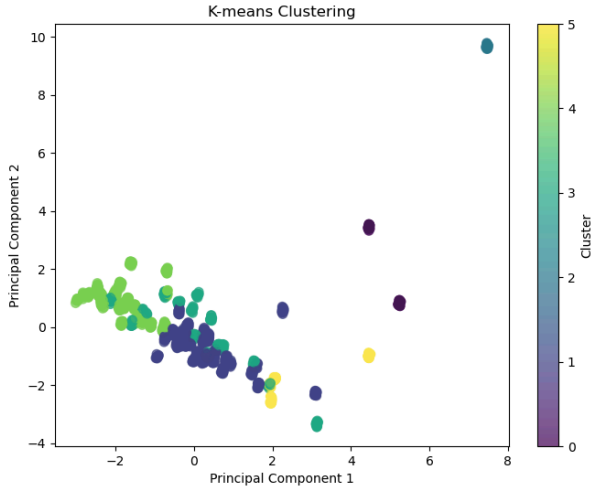

Figure 19: Cluster Analysis (Clusters = 6)

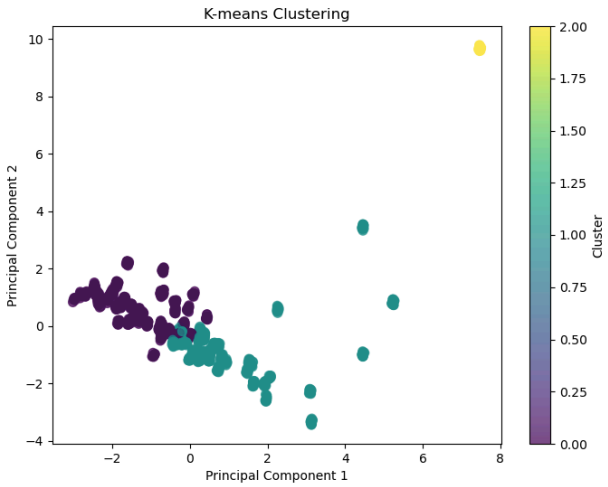

Figure 20: Cluster Analysis (Clusters = 3)

Section 6: Modeling Methodology, Results, and Analysis

As this project is focused on classifying whether a bolt will fail or not and the data set includes a relatively high dimensionality of features, the Machine Learning models selected focus on optimality for classification tasks and high dimensioned data. In the sections to follow, the methodology of splitting the data for modelling using Random Forests and Decision Trees will be described.

To understand each model's performance, a Confusion Matrix is generated to visualize True and False Positives as well as True and False Negatives when comparing the predicted classifications in relation to the actual classifications. Using the data in the Confusion Matrix, a Classification Report is generated to further understand the model's prediction performance. Within the Classification Report, the following records are calculated as defined by the Scikit-YB developers[5]:

- **Precision (Exactness):** Ratio of True Positives to the Sum of True and False Positives
- **Recall (Completeness):** Ratio of True Positives to the Sum of True Positives and False Negatives
- **F1-Score (Mean of Precision and Recall):** Used to compare classification Model's Performance

Splitting the Dataset for Model Implementation

Using the train_test_split function from the Scikit-Learn package, the prepared data is split into training and testing datasets, where the Fracture feature is the Target Variable. To ensure reproducibility, the random state variable is activated so there is a proper distribution of data between the train and test datasets for both the features and target variables. As the target variable theoretically represents a maximum of 100 items of the 1100 item dataset (9.0%), a slightly increased split of the data is used so 70% of the data is in the training dataset and 30% is in the testing dataset.

Model One: Random Forests

For the first model, a random forest using a balanced weight distribution over multiple trees is trained using the Feature and Target variables split from the larger dataset. With a random forest algorithm, the model will be able to handle the high dimensionality as it is immune to dataset features by reducing the overall feature space[6]. To reduce any potential of training bias in the model, a Stratified K-Folds with 10 splits in the data is utilized. In comparison to a standard K-Folds Validation, the Stratified K-Folds works well with high variation in the dataset as the algorithm ensures a representative distribution of all features[7].

After model training, the features test dataset is executed to generate predictors that will be used to visualize the accuracy of the model when compared to the target variable test dataset.

From a performance perspective, each fold of the Stratified K-Folds Validation generates an accuracy of 1.0, thus the total accuracy of the model is 1.0. A confusion matrix was generated using the predicted target variables in comparison to the actual target variables from the dataset to confirm the accuracy, as shown in **Table 4** below.

Table 4: Random Forests Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	316	0
Actual Negative	0	14

From the Confusion Matrix, the conclusion can be drawn that the model performed well, with all 316 Actual Positives being identified through the prediction and are classified as True Positives. Additionally, all 14 Actual Negatives were identified through prediction and are classified as True Negatives. Conversely, there are no False Positives or False Negatives.

Using the Confusion Matrix data, a Classification Report is generated to further analyze the model's ability to classify the target variable based on the features. As depicted in **Table 5**, the model's Precision, Recall, and F1-Score are all calculated.

Table 5: Random Forests Classification Report

	Precision	**Recall**	**F1-Score**	**Support**
True Positives	1.0	1.0	1.0	316
True Negatives	1.0	1.0	1.0	14

Model Two: Decision Trees

Although a somewhat simpler approach to a classification problem, a Decision Tree algorithm divides the dataset into small subsets, similar to Random Forests, but focuses on one tree compared to many, where each branch represents a feature-based decision, and the leaf is the outcome of that decision. Unlike a Random Forest algorithm, a Decision Tree algorithm can be prone to overfitting with highly complex datasets. Given that the dataset in this project has been dimensionally reduced and prepared, the Decision Tree algorithm should not stumble with overfitting and be able to produce an effective prediction.

Using a similar process as performed with the Random Forest algorithm, a balanced weight distribution over multiple branches is trained using the Feature and Target variables split from the larger dataset. To reduce any potential further training bias in the model, a Stratified K-Folds with 10 splits in the data is

utilized. After model training, the features test dataset is executed to generate predictors that will be used to visualize the accuracy of the model when compared to the target variable test dataset.

From a performance perspective, each fold of the Stratified K-Folds Validation generates an accuracy of 1.0, thus the total accuracy of the model is 1.0. A confusion matrix was generated using the predicted target variables in comparison to the actual target variables from the dataset to confirm the accuracy, as shown in **Table 6** below.

Table 6: Decision Trees Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	316	0
Actual Negative	0	14

From the Confusion Matrix, the conclusion can be drawn that the model performed well, with all 316 Actual Positives being identified through the prediction and are classified as True Positives. Additionally, all 14 Actual Negatives were identified through prediction and are classified as True Negatives. Conversely, there are no False Positives or False Negatives.

Using the Confusion Matrix data, a Classification Report is generated to further analyze the model's ability to classify the target variable based on the features. As depicted in **Table 7**, the model's Precision, Recall, and F1-Score are all calculated.

Table 7: Decision Trees Classification Report

	Precision	**Recall**	**F1-Score**	**Support**
True Positives	1.0	1.0	1.0	316
True Negatives	1.0	1.0	1.0	14

Section 7: Analysis of Results

With an Accuracy of 1.0, both the Random Forest and Decision Tree Algorithms excelled at prediction classification of bolt failure based on the training and test data. This is further confirmed with an $R^2 = 1.0$ and Mean Squared Error (MSE) = 0.0 for both algorithms. Additionally, given the perfect scores of 1.0 across the board in the model specific Classification Reports, the models show the ability to perform precisely in predicting bolt failures as well as bolt non-failures.

Knowing that the models execute at a high level for predictions, the next logical step is to evaluate what is learned about the criticality of features that drive these accurate predictions. Using the "feature_importances_" variable in the Scikit-Learn packages, we can plot the overall impact, or importance, of each feature on the resultant model prediction for both Random Forest and Decision Tree

models. As described by Stacy Ronaghan in 'Towards Data Science', "For classification, {Random Forest models} use Gini impurity by default but offer Entropy as an alternative…" where Gini impurity is "calculated as the decrease in node impurity weighted by the probability of reaching that node. The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples. The higher the value the more important the feature."[8]

Represented in **Figure 21** and **Figure 22**, for Random Forest and Decision Tree respectively, the pareto chart shows the contribution of each feature to the prediction accuracy as an overall percentage of importance.

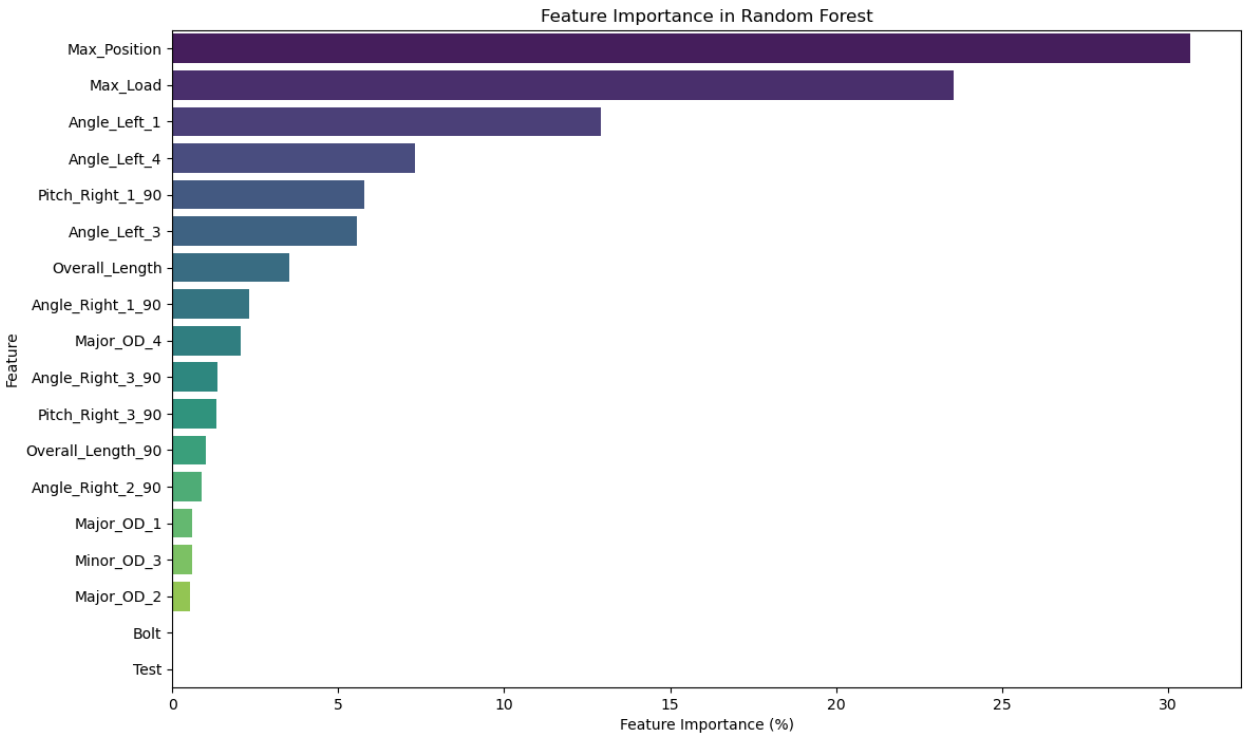

Figure 21: Feature Importance in Random Forest Model

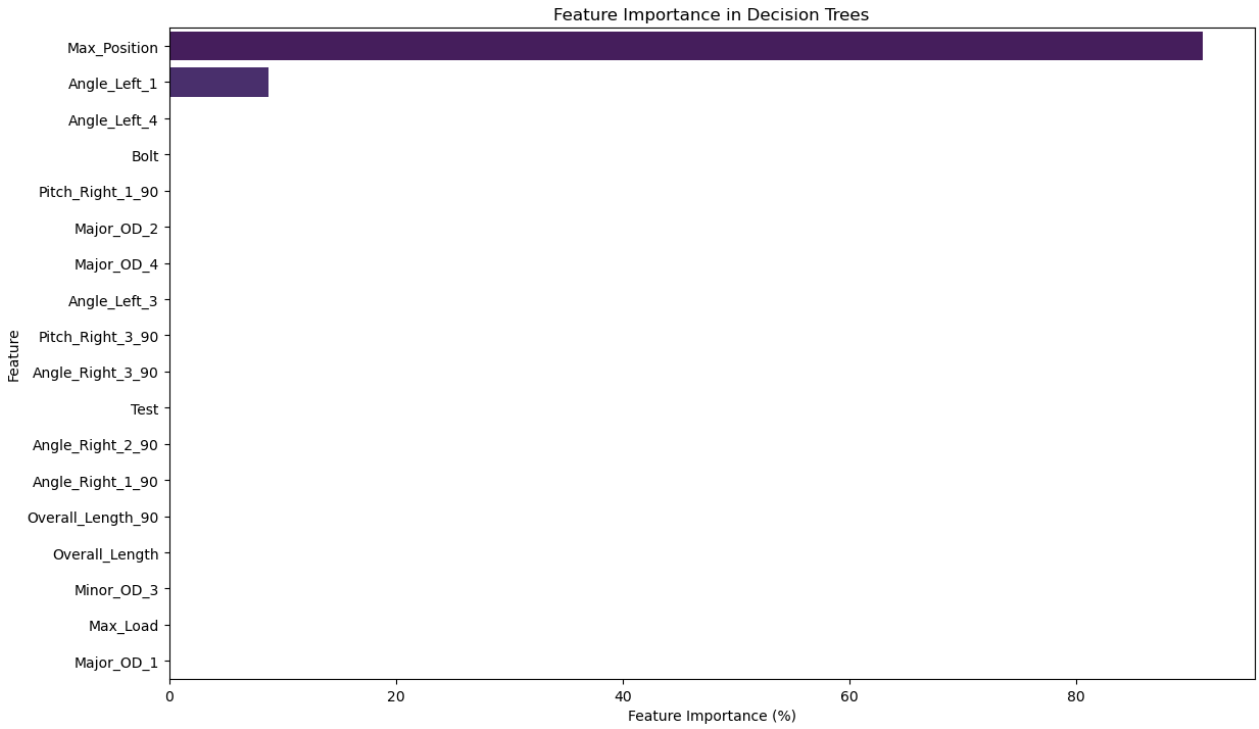

Figure 22: Feature Importance in Decision Tree Model

In reviewing the Importance Graph for the Random Forest Model **(Figure 21),** there is a relatively high percentage of impact to model performance from the Max Position (~30%) and Max Load (~24%) observed by the bolt in the tensile testing machine, with a reduced percentage impact by the dimensional measurements such as Left Thread Angle 1 (~14%) and Left Thread Angle 4 (~8%). Given the ability for a Random Forest model to generate multiple decision trees based on subsets of data, the model can separate features effectively to understand how each feature impacts the resultant prediction.

Conversely, but similar at the same time, the Importance Graph for the Decision Tree Model **(Figure 22)** shows extremely high importance percentage to the Max Position (~95%) observed by the bolt in the tensile testing machine, with a much smaller percentage impact by the Left Thread Angle 1 (~15%) dimensional measurement. As the decision tree model generates a single tree with multiple branches and is generally a less complex model compared to a Random Forest model, the Decision Tree can successfully predict if the bolt will fail or not but achieves this result by focusing predominately on the maximum position achieved by the bolt in the tensile testing machine.

For a professional Material Scientist, these results may not be surprising. Given that the 3D Printed 1” -4 ACME Threaded Bolts are produced out of Nylon 12 powder material with a specified elongation at

break percentage of 4%, the material is excessively brittle compared to other materials and will not hold material deformation as a more ductile material would, similar to the example in **Figure 2** of Section 1 in this report. When connecting the dots between the material's inherent properties and the feature importance being strongly skewed to Tensile Load (Max Load) and Elongation (Max Position), both models were able to correctly identify the most critical features specific to these Physical Twins.

Continuing, as this project is also focused on the ability to use dimensional inspection data to drive failure predictions, both models successfully identified the Left Thread Angle 1 as being the most critical dimensional measurement. As this dimension is positioned at the very first thread, where the maximum load is experienced when the tensile machine pushes down on the threaded bolt assembly, it is established that this thread sees deformation unlike the rest of the dimensional features. Given that these features are generally complex and non-linear when compared to each other, the Decision Tree Model is not able to, as the Random Forest Model can, use these features to drive predictions when a dominant feature such as Max Position is present.

Section 8: Deliverables

There were three key objectives for this project, all focused on developing a robust process of implementing a Digital Twin that can be used to model future performance of the Physical Twin using robust data streams.

For the first deliverable, a 3D CAD model of a 1"- 4 ACME Bolt was designed in Solidworks with a feature-based definition Digital Twin implemented in Azure Digital Twins portal. Using critical dimensions identified in the 3D CAD model and critical mechanical features output from the Tensile Test, a unique Digital Twin entity was generated in Azure Digital Twins for each Physical Twin. Additionally, using the 3D CAD Solidworks model of the CyberGage 360 Laser Scanning unit, a Digital Twin of the measurement system was implemented to demonstrate usage to predetermine inspection processes prior to implementation in a production environment.

Secondly, a data stream was established to ingest dimensional data from the inspection process of the bolt using the CyberGage360 as well as from the Tensile Tests performed on the bolts using the Tensile Testing machine. As the data structures of the data frame output from these machines are not similar, data processing had to occur to provide single unique values for each critical dimension and mechanical feature of each bolt per test cycle. This data is then ingested into the Azure Digital Twin database.

Lastly, using the data in held within the Digital Twin, multiple machine learning models were developed to predict if a bolt will mechanically fail given the historical usage the bolt has experienced. Given the classification problem type, Random Forest and Decision Tree models were trained and tested using the dataset with high accuracy. As a result of this, not only is it determined if the bolt would fail or not, but an understanding is gathered as to which features impact the overall performance of the bolt.

Section 9: References

1. *CyberGage®360* (2023) *CyberOptics*. Available at: https://www.cyberoptics.com/products/cybergage360/ (Accessed: 16 February 2024).

2. (No date) *Cybergage360 accuracy specification - cyberoptics*. Available at: http://www.cyberoptics.com/download/industrial-metrology/powered-by-mrs/CyberGage360_Accuracy_Specification_2017.pdf (Accessed: 29 January 2024).

3. *CyberGage®360* (2019) *Laser Design*. Available at: https://www.laserdesign.com/products/CyberGage360/ (Accessed: 25 April 2024).

4. Azure (no date) *Opendigitaltwins-dtdl/DTDL/V2/dtdlv2.md at master · Azure/opendigitaltwins-DTDL*, *GitHub*. Available at: https://github.com/Azure/opendigitaltwins-dtdl/blob/master/DTDL/v2/dtdlv2.md (Accessed: 12 February 2024).

5. *Classification report𝔍* (no date) *Classification Report - Yellowbrick v1.5 documentation*. Available at: https://www.scikit-yb.org/en/latest/api/classifier/classification_report.html (Accessed: February 2024).

6. Simplilearn (2023) *Random Forest algorithm*, *Simplilearn.com*. Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm (Accessed: 19 March 2024).

7. *Sklearn.model_selection.Stratifiedkfold* (no date) *scikit*. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html (Accessed: 05 April 2024).

8. Ronaghan, S. (2019) *The mathematics of decision trees, random forest and feature importance in Scikit-learn and Spark*, *Medium*. Available at: https://medium.com/@srnghn/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3 (Accessed: 25 April 2024).

Section 10: Self-Assessment

Having worked in Manufacturing Operations as a Manufacturing Process Engineer the entirety of my 10-year career, I have approached each course within the DSA program, including this project, from the lens of applicability to these professional experiences. Given that industry is well into the Industry 4.0 transition, there is an immense about of content generated on focusing data driven improvements and decision making from a theoretical perspective, with a significantly smaller percentage of work focused on true application within this space. To this point, I feel honored and lucky to have had the opportunity to produce work showing the implementation of a Digital Twin and how these IoT models can be used to improve manufacturing product quality and reliability.

Dr. Beattie and Dr. Nicholson have repetitively said that the most time-consuming aspect in any data science project will be focused on preparing and evaluating data prior to performing modelling. Given that this project had an aspect of generating real-world data to use in Machine Learning models, I had to dig deep into learnings from multiple classes to effectively prepare the data to be in a usable state. Additionally, as I was generating data using the CyberGage 360 and Tensile Tester, given that the bolt I produced was manufactured using a generally brittle material, I did not see any data correlations during the data generation phase of this project. It was not until implementing data evaluation methods that correlations and relationships were visualized.

From a self-learning perspective, this project used the Azure Digital Twins Environment which is not a tool presented in the DSA program. Due to this, extensive work had to be performed to understand the functionality of this tool and most importantly, how to establish data streams to ingest acquired data into the environment. This is ultimately extremely challenging as there are security protocols in place to protect Azure based data that needed to be understood and overcome.