TITLE: NOVELTY DETECTION FOR PREDICTIVE MAINTENANCE

by

Michael F Finch

August, 2020

Director of Thesis: Professor Nassah Tabrizi

Major Department: Computer Science

Since the advent of Industry 4. 0 significant research has been conducted to apply machine learning to the vast array of Internet of Things (IoT) data produced by Industrial Machines. One such topic is to Predictive Maintenance. Unlike some other machine learning domains such as NLP and computer vision, Predictive Maintenance is a relatively new area of focus. Most of the published work demonstrates the effectiveness of supervised classification for predictive maintenance. Some of the challenges highlighted in the literature are the cost and difficulty of obtaining labelled samples for training. Novelty detection is a branch of machine learning that after being trained on normal operations detects if new data comes from the same process or is different, eliminating the requirement to label data points. This thesis applies novelty detection to both a public data set and one that was specifically collected to demonstrate a its application to predictive maintenance. The Local Optimization Factor showed better performance than a One-Class SVM on the public data. It was then applied to data from a 3-D printer and was able to detect faults it had not been trained on showing a slight lift from a random classifier.

NOVELTY DETECTION FOR PREDICTIVE MAINTENANCE

A Thesis

Presented to the Faculty of the Department of Computer Science

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Master's of Science in Computer Science

by

Michael F Finch

August, 2020

NOVELTY DETECTION FOR PREDICTIVE MAINTENANCE

by

Michael F. Finch

APPROVED BY:

DIRECTOR OF
THESIS: ____
M. N. H. Tabrizi, PhD

COMMITTEE MEMBER: ______________________________________
Maral Azizi, PhD

COMMITTEE MEMBER: ______________________________________
Venkat Gudivada, PhD

CHAIR OF THE DEPARTMENT
OF COMPUTER SCIENCE: ______________________________
Venkat Guidivada, PhD

DEAN OF THE
GRADUATE SCHOOL: ______________________________________
Paul J. Gemperline, PhD

ACKNOWLEDGEMENTS

I would like to thank Jabil Inc for providing access to the 3D Printers and other resources to complete this research. I would also like specifically thank Timothy Langston and Nathan Cantwell from the Jabil Blue Sky facility for their patience and assistance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Since the concept of Industry 4.0 was first released at the Hanover Fair in 2011 [1] the potential of the Industrial Internet of Things (IIoT), Big Data, and Machine Learning (ML) to revolutionize manufacturing has been the subject of both the trade press [2] and research efforts. One of the fields IIoT was expected to impact was predictive maintenance (PredM) [3]. Maintenance itself is not a new idea, there is a whole field of reliability engineering focused on methods to optimize the life of machinery. Industrial equipment is designed to have a long lifetime and relies on extensive maintenance to keep it operational. The simplest maintenance scheme, Run to Failure (RTF), has been around for as long as there have been machines. This scheme is still used today in areas where the impact of failure is low. Preventative Maintenance (PrevM) was designed to reduce this cost. It uses a fixed cycle of maintenance based on time, operating hours, or cycles to do maintenance at a convenient time. This increases the cost of doing maintenance but improves the overall operational budget with fewer unplanned failures. Reliability Engineering has extensively studied and optimized PrevM, but unplanned failures still occur. This led to the development of PredM, sometimes referred to as Condition Based Maintenance[4]. Condition based maintenance started out as a simple visual inspection when doing other maintenance, replacing parts that have shown wear. The next step in PredM is using a periodic measurement, an automotive example of this is measuring the brake pads and replacing below a certain thickness. These two levels of PredM are firmly in the reliability engineering field. The third level is using sensors to provide continuous data and scheduling maintenance based on a sensor exceeding a rule. [5] Modern Industrial Equipment like Wind

Turbines, Semi-conductor Manufacturing Equipment, and Aircraft Engines are full of sensors generating thousands of data points every few seconds. Modern Computer Science methods are required to gather and analyze this data, and to use it to predict that a failure will occur in the future. Using ML to find patterns in this data has the potential to not only reduce the number of unplanned failures, but also the cost of performing unneeded maintenance. However, a study by IIoT service provider PTC in 2017 showed that only 3% of IoT projects were for a PredM use case [6]. This thesis will review the current state of the art in applying ML to PredM and apply it to Additive Manufacturing. Additive Manufacturing, commonly known as 3D printing is a rapidly growing field in manufacturing. Parts are built up layer by layer in shapes that cannot be economically manufactured by other means [7]. The manufacturing process takes hours to make the part, and additional hours to allow it to cool and be removed. The resulting manufacturing cycle can be measured in hours, or days, and is typically unattended. This provides an additional value to PredM in additive manufacturing, to detect the failure as it occurs, preventing the waste of machine time and improving overall equipment effectiveness (OEE).

As will be covered in more detail, typical machine learning implementations in PredM are classification problems, which require labelled data. Labelled data can be hard and expensive to obtain. This thesis addresses that problem by using novelty detection. Two novelty detection methods are applied to a public data set, and then one of those is applied to data collected from a 3D printer farm. In both cases the novelty detection methods detect failures that were not available in the training set.

The contributions of this thesis are:

1) The survey of machine learning in predictive maintenance, which will be submitted for publication [8].
2) An implementation of Novelty Detection Methods to find faults in data sets designed for testing and evaluating predictive maintenance models.
3) A demonstration that Local Optimizer Factor was capable of detecting faults in a real-world 3D printer data set.

The thesis is structured as follows. Chapter 2 describes the current state of the art in using ML for PredM, Chapter 3 covers the 3D printer data set, describing the exploratory analysis and data cleaning methods used in this work, Chapter 4 describes the modelling and results and Chapter 5 provides a summary of the results and recommendations for future work.

CHAPTER 2: RELATED WORKS

The starting point for this research was to conduct a survey of existing research in this field. After summarizing the results of the survey, we will review the methods we used to develop and analyze the survey. Many authors have demonstrated that ML methods can be used in the field of PredM. Research to date has focused on traditional ML classification methods. Many different industries are moving to take advantage of Industry 4.0, and a wide variety were represented in this survey. A theme that arose through this research is that more attention needs to be paid to evaluating a model in this field. One avenue for investigation includes using unsupervised or semi-supervised learning methods to avoid the need for large numbers of labelled examples.

2.1 Survey Methodology

The initial review of related works was a keyword search of Google Scholar for “machine-learning” in PredM from 2008 to 2018. As shown in Figure 2.1, there was a major jump in interest in the topic starting in 2014.

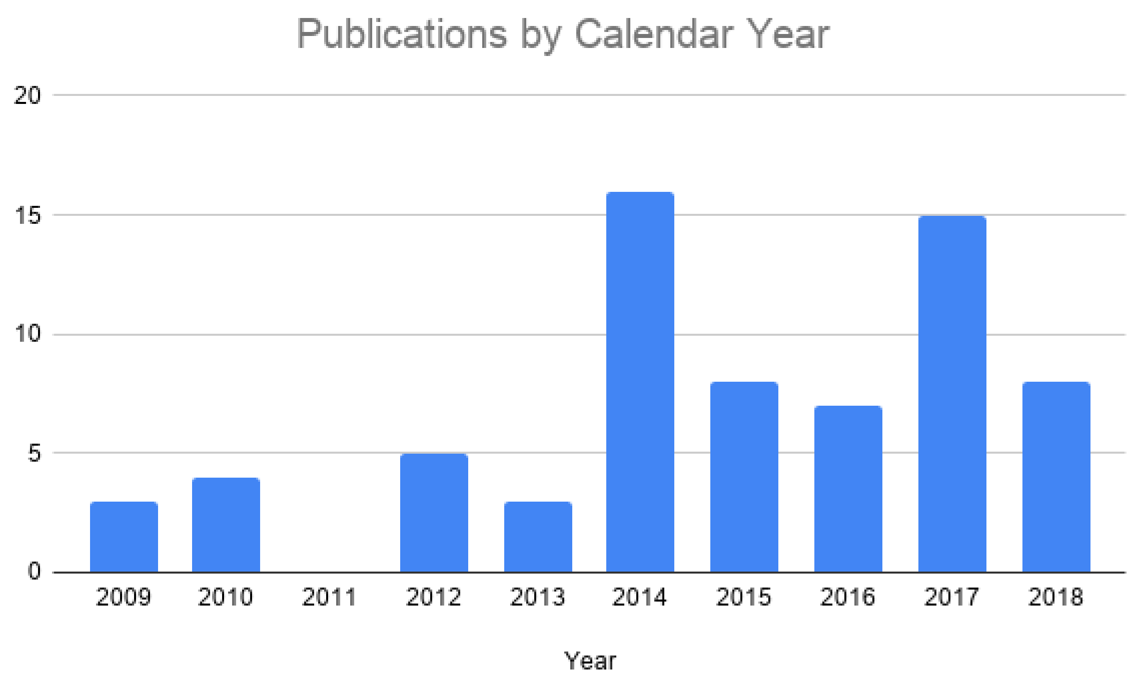

Figure 2.1 Publications for Machine Learning Predictive Maintenance in Google Scholar

Based on this initial screening, this research focused on the period from 2014 onward where there was significant activity in the field. IEEE, ACM, and DBLP were searched for additional papers, and the entire list was scrubbed to only include journal and conference papers. Each paper was read to confirm relevance to the topic, resulting in sixty-two papers over six years. While there is still an increase over time, it is not as sharp as the jump in 2014, see Figure 2.2 The decrease in 2019 may be due in part to the collection being gathered in the fall of 2019 and not representing a full year.

Figure 2.2 Relevant Papers for Machine Learning Predictive Maintenance from 2014 to 2019.

The first investigation was to look at where the papers were published and what the authors thought the contribution of the work was. There was no pattern of publication source with all sources that published multiple papers on this topic listed in Table 2.1, the remaining 51 papers came from 51 unique publications.

Table 2.1 Publications with Multiple Papers

PUBLICATION	NUMBER OF PUBLICATIONS	PUBLICATIONS
IEEE Conference on Big Data (Big Data)	3	[9], [10], [11]
IEEE Conference on Emerging Technologies and Factory Automation (ETFA)	2	[12], [13]
IEEE Transactions on Industrial Electronics	2	[14], [15]
Procedia Computer Science	2	[16], [17]

These papers were then categorized by the methods, data set, and industry represented in the paper. These topics will be explored in more detail in the coming sections. The final area reviewed was the purpose of the paper. Most of the papers were demonstrations that ML could be applied to the problem of PredM, including a documentation of a live demonstration using a neural network to detect motor vibration[18]. The remaining categories of method, dataset and architecture are shown in Figure 2.3.

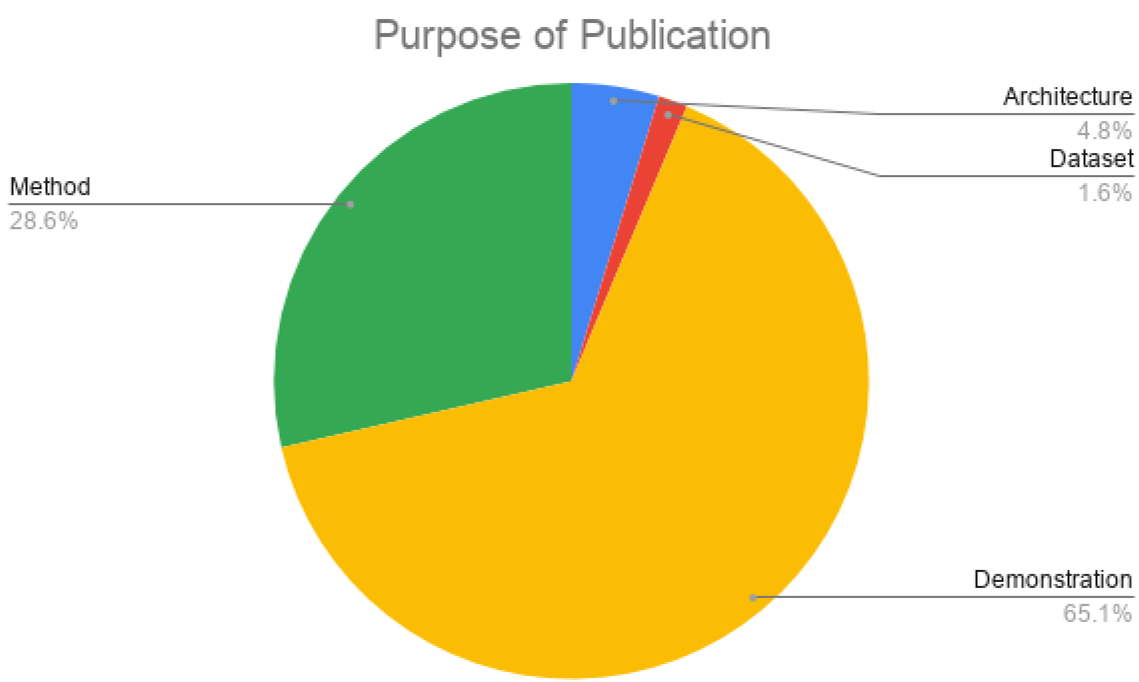

Figure 2.3 Purpose of Publication

2.2 Methods

The first dimension to explore is the methods used for ML in the PredM field. A majority of the papers published during these six years used traditional ML methods (ML) over Deep Learning, see Figure 2.4. Deep Learning started to receive more attention recently as the number of deep learning papers approached ML in 2017 and exceeded it 2019. The total exceeds the number of papers in some years, as some papers compared the effectiveness of multiple models.

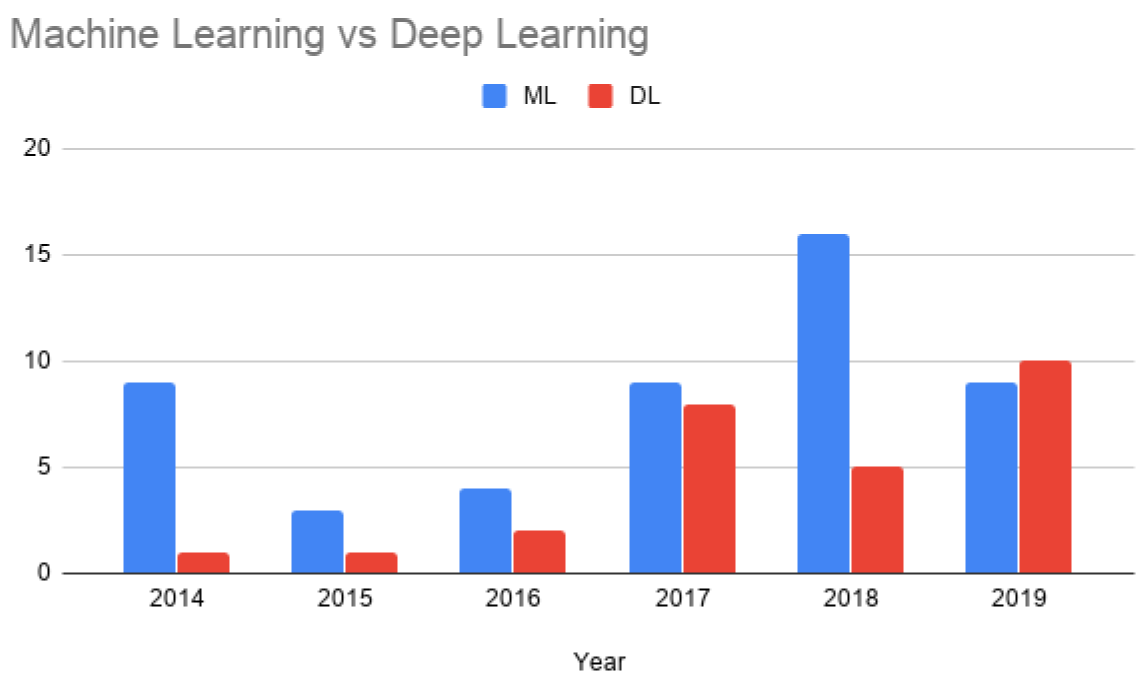

Figure 2.4 Machine Learning vs Deep Learning

Traditional ML methods were used by most of the papers in this study. Some of these demonstrated the effectiveness of a single method, with 3 papers demonstrating the use of a random forest (RF) [19] [20] [21]. Other examples include using a Support Vector Machine (SVM) to do classification on rail cars [22]. There were also comparisons of multiple methods, showing that a Gradient Boosted Tree (GBT) and RF outperforming an SVM and Linear Regression (LR) [23] on a public hard drive failure data set [24], and similar results on vending machines [25]. Logistic Regression (LogR) and RF were found to perform similarly [26] on a public data set for heating ventilation and air-conditioning (HVAC) machines [27].

The reasons for the low penetration of deep learning in PredM has not been studied. Potentially it may be due to the audience, the sponsors of commercially successful models in the NLP and video recognition models tend to be tech company executives, and the front-line operators are software engineers and Product Managers. The PredM sponsors tend to be Industrial Operations Executives, and the actual users are reliability engineers. Their acceptance of models might be higher if they can interpret the model, and not required to treat it as a black box [28]. It may also be that traditional methods such as RF, kNN and SVM were able to achieve satisfactory results. As discussed later in this survey, the PredM function is driven by cost to a greater extent than some other domains, and the additional cost in gathering labeled data, training, and operating a deep learning model may not drive enough benefit.

Figure 2.5 breaks down the deep learning models in more detail. Most of deep learning models were traditional artificial neural networks (ANN), which many papers referred to as a Multi-layer Perceptron (MLP). Some examples include the use a Deep Belief Network to do classification on machining centers [14], and Extreme Learning Machines to conduct regression on bearing data [29], and classification on Wind Turbines [30]. There was a new deep learning method, Anomaly Detection-based Power Saving (ADEPOS), tailored specifically for PredM. This method increased the complexity of the ANN as anomalies were detected to improve the accuracy of the prediction while consuming less power early in the life of the component to minimize the compute power required [31].

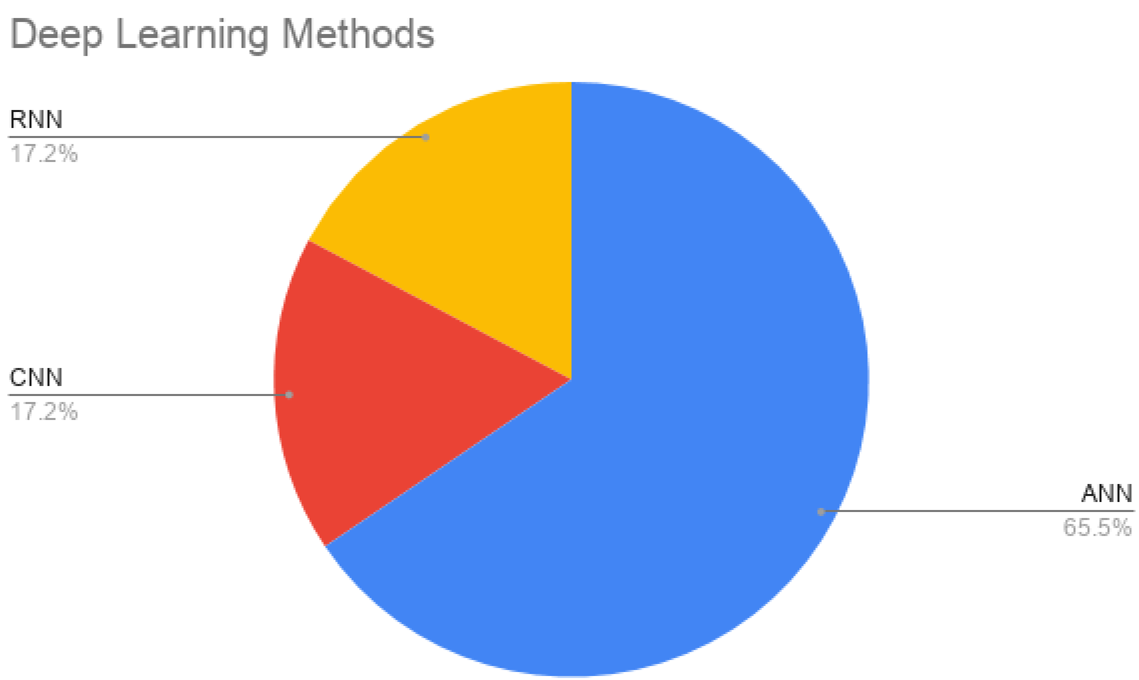

Figure 2.5 Deep Learning Methods

There was also an example of a new method using Recurrent Neural Networks (RNN) for machine health monitoring. The method used a Long Short-Term Memory (LSTM) model for modeling the incoming time series data from the assets. It was then fed into an ANN for feature learning, and that output was sent to a survival model [32]. The model was tested both on field data from heavy trucks and an open source hard drive data set [24]. An additional exploration of RNNs was done on 3 data sets this time using a local feature-based gated recurrent unit network (LFGRU) [15]. Although Convolutional Neural Networks (CNN) are typically associated with the computer vision domain, there were examples of their utility in PredM, such as this demonstration of using a CNN to detect a fault in a PV panel [33].

Multiple papers did not restrict themselves to just deep learning or just traditional ML. Several of them compared deep learning methods with traditional ML. A comparison of multiple methods on a public data set of machining data determined that RF provided better regression results than an ANN [34]. While a separate study on industrial welding robots determined that a CNN outperformed Extremely Randomized Trees, kNN, and Survival Analysis. Of note is the conversion of sensor data to a grey-scale image prior to using the CNN [16]. Still another study showed an ANN outperforming several ML models on a classification task [35]. A study on heating ventilation and air-conditioning (HVAC) systems compared multiple ML and deep learning models on two different fault classes, again determining that a RF model had the best performance on both faults [36]. Finally, both Gradient Boosting (GB) and an ANN were demonstrated on motor encoder data [37].

There were also several examples of combining traditional and deep learning. An ensemble model of GB, RF and an ANN were used to classify faults in semiconductor manufacturing equipment [38]. Deep learning and ML methods were also used in series, connecting the output of a CNN to the input of an SVM [39].

Since the data source is a collection of sensor data taken from machines over time, it stands to reason that time-series models also appear in the literature. Multiple ML models were used as an input to an Auto-regressive Moving Average (ARMA) model to predict the remaining useful life of aircraft bleed valves. Compared to a traditional reliability life usage model, the only model to underperform was an ANN, while four traditional ML models had better results, the best was an SVM [40]. An Auto-regressive Integrated Moving Average (ARIMA) method was used as a pre-processing step to train sensor data, and then feed it into a PCA for dimensionality reduction, that output was provided to an SVM to classify events [41].

More pronounced than the use of traditional models is the preference for supervised learning. As shown in Figure 2.6, 90% of paper used supervised learning, with only token examples of unsupervised and semi-supervised learning. The single semi-supervised paper compared the performance of 3 semi-supervised anomaly detection algorithms to 5 unsupervised clustering algorithms[9]. They used the F1-Score, the harmonic mean of Precision and Recall, to compare methods. They saw a better F1 score on the anomaly detection (0.78 - 0.89) vs the clustering (< 0.70). However, the same paper also compared the unsupervised/semi-supervised models to supervised classifiers. The supervised classifiers had a significantly higher F1 score (0.98 for Random Forest vs 0.89 for Isolation Forest).

Strong arguments for the value of unsupervised learning is the difficulty in getting labelled data, and its ability to detect faults that are not present in the training set. This was the motivation for a comparison of multiple unsupervised models on vibration data from HVAC equipment [42]. Other examples showed use of a prefix tree [12], one class SVM [43] and AdaBoost [44]. There was also a single example of an unsupervised ANN [18].

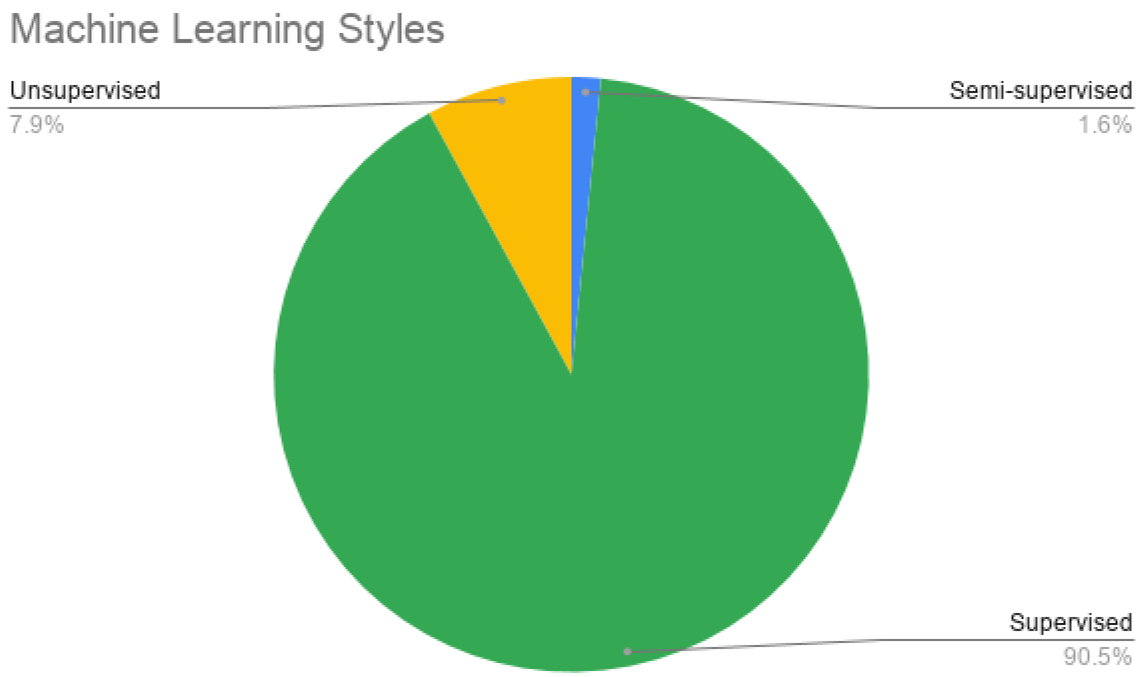

Figure 2.6 Comparison of Machine Learning Styles

As shown in Figure 2.7, there is also a slight preference for classification methods. Classification methods provide a signal for the maintenance personnel, with either a “repair now” or “normal” result. These results can be tied into an Enterprise Asset Management program to generate a work order, order a spare part, or even stop a process. A regression method can be used to provide more information to the maintenance engineer by reporting an estimated time to failure or remaining useful life. This can be used to schedule maintenance at an appropriate time, which can be of significant value.

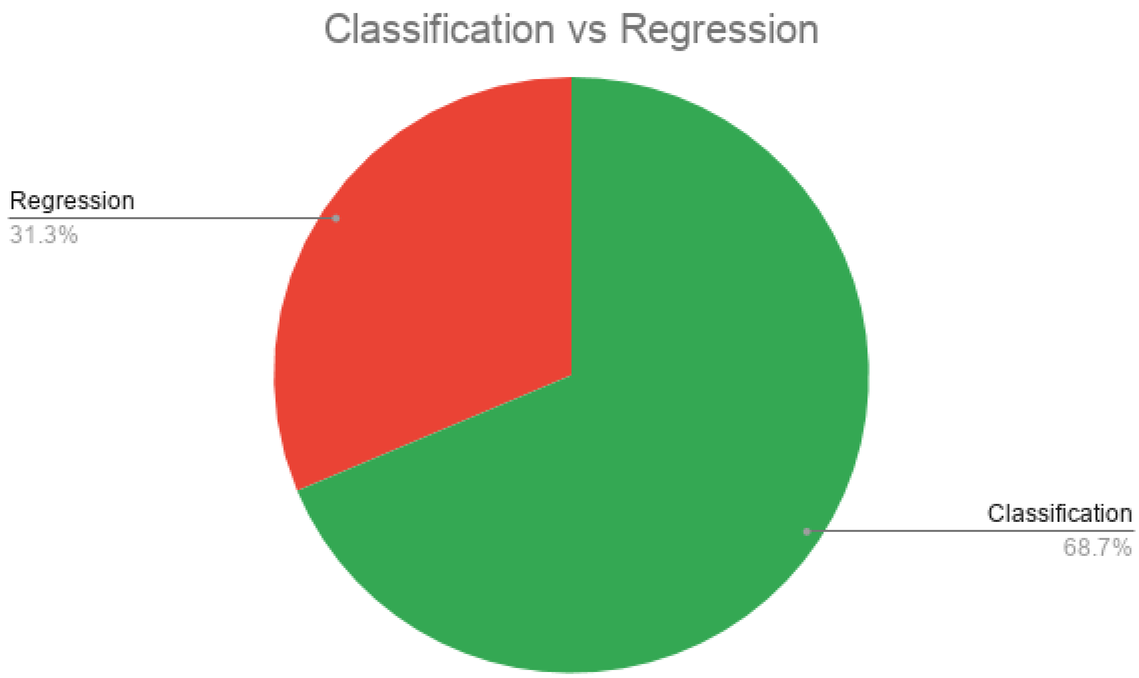

Figure 2.7 Comparison of Classification vs Regression

One cluster of regression methods was based on the Nasa Engine Set [45] which was intended for that purpose. Authors provided results using an Artificial Neural Network [46], a Recursive Neural Network [47], and a Support Vector Machine [48]. Some research involved both classification and regression, sometime for two separate problems [49], but one paper compared the two with classification achieved a better MSE than regression on the same data set [50].

There were a relatively few papers that created new methods ML. One area where a cluster of papers addressed some non-standard techniques was in the evaluation of a model. Unlike some other fields where ML has made major inroads, the cost function for a PredM model can be easily calculated in dollars. Failure to detect a needed maintenance will result in a production stoppage, which can result in huge costs, while replacing a part unnecessarily also has costs in maintenance effort, and the loss of useable functionality of the replaced part. One of the most detailed examples of showed that by assigning real costs to a PredM model the behavior of the decision tree was dramatically different than if the goal were to maximize the F1 score [51]. The optimal model using a traditional method of maximizing F1 score would result in a higher cost of maintenance than not doing any predictions at all. False positive reduction was a factor in evaluating the model for multiple papers. The cost of deploying a field service engineer to a machine on a customer's premise was one reason [52]. Another reason provided was the limited maintenance budget. When tearing up streets to replace water pipes, there is a limit to the number of repairs that can be made in one season, only the highest confidence predictions were evaluated, since those were the only ones that could be acted on economically [53]. In addition to the economic cost, a high false positive rate could impact operator confidence in the algorithm [54]. While most papers focused on the cost of downtime and the waste of replacing parts with remaining useful life, two authors also looked at the computing cost and energy involved in training and operating the predictive models. [31] [17]

2.3 Data Sets

ML consists of two areas, an algorithm, and data, of which data may be more important [55]. As shown in Figure 2.8, most authors used proprietary datasets that they gathered from the field, these correlate with the leading purpose of the publications, to demonstrate that the ML methods developed primarily in other fields could be applied to PredM. Some of the data sets encompassing many samples over many years, like this example using an ANN to predict failure within the next year based on oil piping inspections [56], while others implemented a SVM on a single heat exchanger [57]. Regardless of the number of machines, I classified a data set source as field data if the machines were being used in normal operating mode.

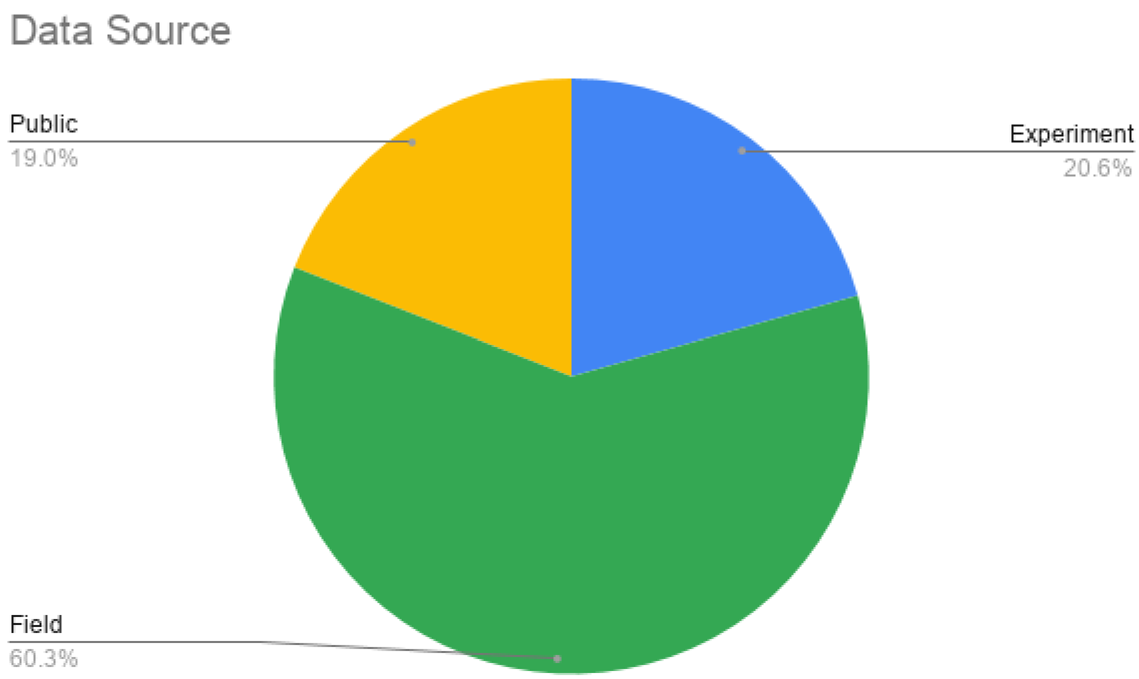

Figure 2.8 Data Sources

One of the issues with applying ML to PredM is the class imbalance, typically a manufacturing operation runs most of the time in a productive state with only a very few instances of a failure. This issue is not unique to PredM, it is also common to medicine and public health. One resolution in the PredM field where there are fewer ethical concerns was to run the machine under controlled conditions that were guaranteed to produce a failure. This advantage was exploited on a single gearbox to develop an SVM classifier [58] and on a set of over 600 bearings to develop an ANN Classifier [59]. These data sets are classified as Experiments.

Public data sets are also used, but less commonly than in other domains. One paper did try to alleviate this by publishing a data set on marine engines that [60] that was subsequently cited by [61]. Some of the other frequently cited datasets were the NASA Bearing [62] and Engine datasets [45]. Table 2.2 shows the representation by industry, the component category includes items like motors and bearings which could be generalized to multiple industries.

Table 2.2 Publications by Industry

INDUSTRY	NUMBER OF PUBLICATIONS	REFERENCES
Transportation	14	[17], [20], [21], [22], [32], [40], [41], [46], [47], [48], [50], [60], [61], [63]
Discrete Manufacturing	10	[9], [11], [14], [16], [34], [64], [65], [66], [67], [68]
Process Manufacturing	9	[12], [13], [35], [38], [51], [54], [69], [70], [71]
Energy	9	[30], [33], [39], [49], [56], [57], [72], [73], [74]
Infrastructure	8	[10], [27], [36], [42], [43], [44], [53], [75]
Component	7	[15], [18], [29], [31], [37], [58], [59]
Other	6	[19], [23], [25], [28], [52], [76]

Within some of these broad categories there were clusters of papers. A significant cluster was 8 of 10 Discrete Manufacturing papers being related to machining. Seven of them used vibration sensors to predict cutter life. The one exception used the existing process data from a machining center as the input to an ANN to generate a regression model [65]. Of the remainder vibration sensors, one created a classification model using an ANN with just the vibration data [14]. The other six paired the vibration sensors with power consumption, temperature, acoustic and/or cutting force sensors. These six were evenly split between ML and deep learning methods, and between regression and classification [18][34][64][66][67][68].

There was another cluster of six papers in Semiconductor manufacturing. These showed a similar mix between classification and regression, however they were predominantly ML. One deep learning model was used as part of an ensemble method [38], and one two were in papers that compared multiple methods. Both got best results from decision tree-based methods [69][70]. The remaining three papers in the semiconductor manufacturing domain were all conducted on field gathered data by the same author [51][71][13].

The final large cluster of papers was in the wind energy sector, with an additional six papers. Unlike the other two clusters the wind turbines were all classification models. This category had the one paper that was primarily about the architecture [74]. It concentrated on how to collect and process the data to generate a real time prediction using RF. This category also had a comparison between a deep learning and an ML model, this time the ANN beat Naïve Bayes (NB) [73]. There was also another paper that used both a deep and ML in series with a CNN feeding into an SVM [39]. There were also two papers the just used a single model, an RF [72] and an ANN [30]. These 5 papers used the data provided by the normal turbine operation. There was also an ensemble model that used vibration data to detect faults in the turbine gearbox bearings. It was not clear if the vibration sensor was an addition to the turbine or part of the installed data collection [74].

This illustrates two different options in the gathering of data to support the ML model. New machines provide an enormous volume of data during operation, and ML is needed to extract more value out of the data. This is typical of the semiconductor and wind turbine papers. In addition to the examples above, there was also two examples of generating a ML model for medical diagnostic imaging machines [28][52]. Older machines do not have either the sensors or connectivity suite that these newer machines have. A detailed study of the challenges of adding sensors and communication systems, gathering data, and analyzing it in a brown field manufacturing environment is provided by [9]. While a similar example for rolling stock and trackside components in a railway system is [63]. One of the most unusual data sets was a PredM model built of sentiment analysis from social media. [76]

As we saw in the Figure 2.1, using Machine Learning for predicative maintenance is a very new field. The research is recent, and only beginning to be implemented in the field. This provides an advantage for early implementations, given the current state of the art in many industries is limited, the threshold for making a material improvement is lower. A comparison of Random Undersampling with AdaBoost to random forest in distribution transformers found the RUSBoost had better recall at 35% to 31%, however this far outperformed the state of the art at 5% [75].

CHAPTER 3: DATA COLLECTION AND PRE-PROCESSING

3.1 Problem Statement

The term Additive Manufacturing was selected to represent multiple technologies including vat polymerization, powder bed fusion, material extrusion and binder jetting to differentiate them from the machining process above where material is subtracted from a blank to create a part. The literature shows some challenges in the additive manufacturing but is focused on higher level challenges like the availability of process simulation tools and the environmental impact where additive manufacturing is different from other industries [77]. We then discussed some of the challenges of the additive manufacturing process with Subject Matter Experts in Jabil's Manufacturing Engineering department working with print farms. They had similar concerns to manufacturing engineers everywhere: yield, OEE, maintenance expenses, etc. They provided examples of several commonly occurring issues in production farms that are hard to predict. These issues have costs in material and machine time and can often cascade to multiple parts causing a more expensive failure. They suggested two that could be easily and safely injected into the machines, a tangled filament, and a failed cooling fan. Our goal for this study was to develop a model that could detect these issues before they occurred allowing an operator to intervene, and either prevent the defective part, or at least be able to clean and restart the printer saving hours of machine time.

This research was conducted using the data provided from a farm of 12 Ultimaker 3 printing machines [78]. Like the wind turbine and semi-conductor manufacturing examples provided above, the Ultimaker machines have embedded sensors to capture process variables. They also have an API which allows you to poll for information from the machines, eliminating much of the barrier to initiating IIoT. A simple python script running on a Raspberry Pi was set up to poll these machines. The data was stored in PostgreSQL an open source relational database, which for ease of setup and maintenance is run as a managed service in Microsoft Azure. Numpy [79] and pandas[80] are the primary Python open source packages used to analyze the data in a Jupyter Notebook, also running in Microsoft Azure. Additional packages cited in text where their use is discussed.

3.2 3D Printer Data

All told there were over one hundred data fields available from the machine. Not all of them were relevant to this task, and for some, although the field was defined, the data value never changed. After selecting values that might be relevant to the task, the data was stored in two tables. The first that stored the static information about the printing jobs: part being printed, material, printer identification, and software version, etc. The second captured data that varied as the part was printed. Because some of this data is proprietary, it has either been obfuscated or removed from this report. A list of the fields included in these two tables are shown in Table 3.1 and Table 3.2. The notes column indicates if any obfuscation was done. All the temperatures were obfuscated by adjusting them by a constant. The same constant was used by all numerical columns, so that differences remained constant. Zero values were not modified to avoid making the constant easy to guess.

Table 3.1 List of fields in the Prints Table

FIELD NAME	DESCRIPTION	NOTES
uuid	unique identifier of each print job	Primary Key for prints, foreign key for data
part_id	identifier of the part being built	hashed to hide actual part name
printer	name of printer	allows identifying differences between individual machines

datetime_started	time when print started	can be used with printer to put prints in order
material1	material in hotend1	
hotend1	id of hotend1	
material2	material in hotend2	
hotend2	id of hotend2	
total_time	time for print job	expected
result	how job completed	"Finished" – job ran to end "Failed" – machine fault that stopped job "Aborted" – machine stopped by operator
datetime_finished	time print ended	in case time between prints is relevant

Table 3.2 List of Fields in Data Table

FIELD NAME	DESCRIPTION	NOTES
timestamp	UTC timestamp of data	
uuid	id of print	
bed_preheating	bed status	boolean
bed_temp	temperature of bed	anonymized by a constant shift applied to all temperatures. Missing values are 0
bed_target	target bed temp	same anonymization
hotend1_temp	temperature of hotend1	same anonymization
hotend1_target	target temp of hotend1	same anonymization
hotend1_max	maximum temp of hotend1	same anonymization
hotend1_material_extruded	amount of material from hotend 1	
hotend1_hot_time	hotend 1 time spent hot	
hotend2_temp	temperature of hotend2	same anonymization
hotend2_target	target temp of hotend2	same anonymization
hotend2_max	maximum temp of hotend2	same anonymization
hotend2_material_extruded	amount of material from hotend 2	
hotend2_hot_time	hotend 2 time spent hot	
state	state of the print cycle	
progress	fractional completion of print job	
time_remaining	time left in the print job	
time_elapsed	time elapsed in the print job	

The first step in looking at the data was to look for missing values. There were very few, just two rows in the prints table that were missing the material and hotend ID for both hotends one and two. The plan to deal with these missing values was to look for other prints of the same partid and populate with the most common data item for those prints, they should all be the same. If there were no other prints of that part, then those rows and the corresponding rows in the data table would need to be dropped.

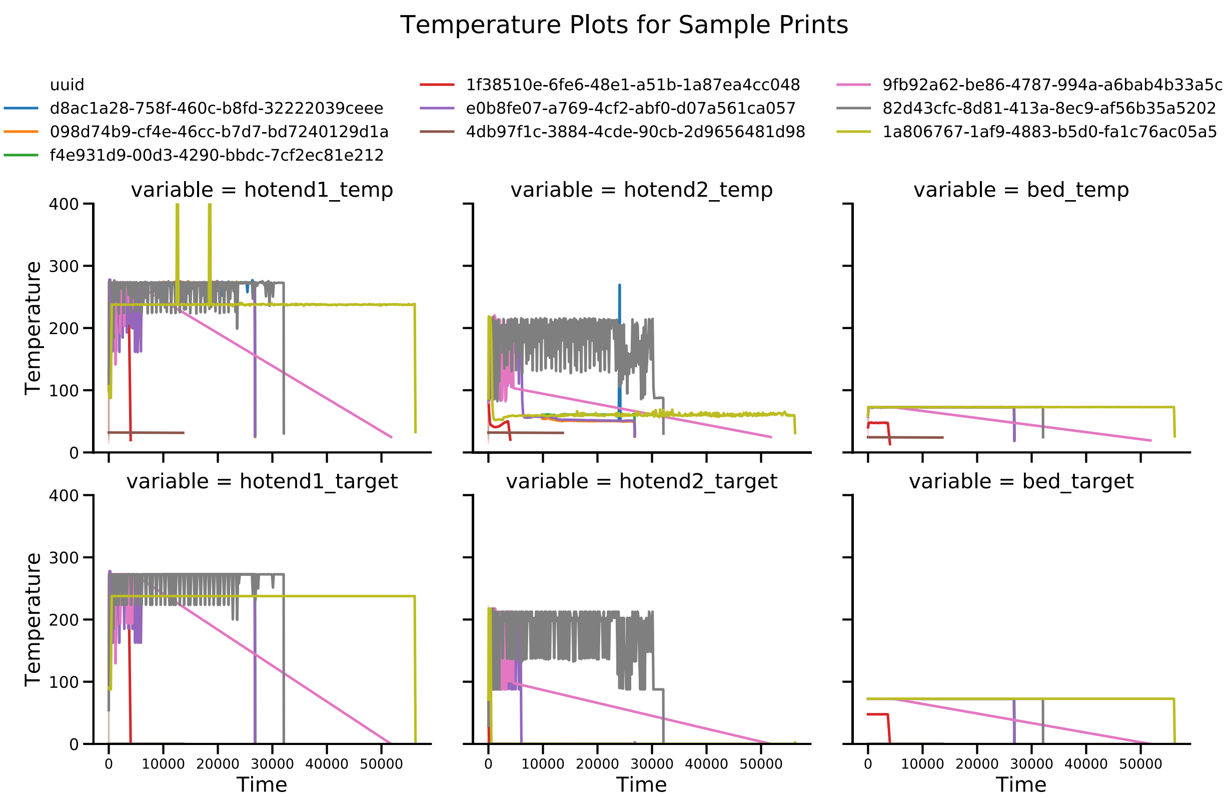

Figure 3.1 Temperature plots of typical 3D Print runs

The next step was to plot the target and actual temperatures from the three heaters as shown in Figure 3.1 using seaborn[81]. There are a couple of interesting artifacts in the plots. The diagonal line for the print with uuid “9fb…” is an artifact where data collection was stopped at 9% complete and did not resume until after the printing had stopped, a delay of approximately 13 hours. There are also some spikes of hotend1_temp and hotend2_temp, where they take a large jump for one reading. Two of them jump off scale in the top left chart, and there is one in the second chart. These appear to be sensor anomalies. This chart and all others unless specifically noted was created with matplotlib [82].

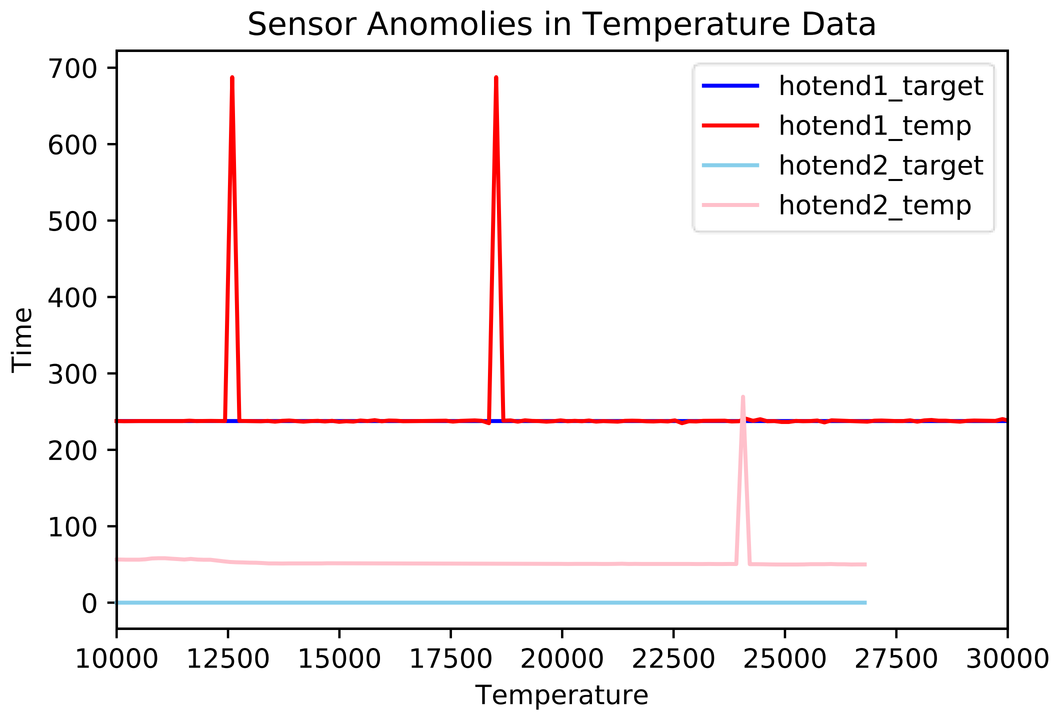

Figure 3.2 Sensor Anomalies in Temperature Data

The first attempt to filter them out looked for items that where the target temp was significantly different than the actual temp this selected the sensor anomalies in

Figure 3.2, but it also selected several places where the target temperature changed more rapidly than the actual temperature could follow Figure 3.3. To eliminate the spikes, if a point's value was substantially higher than both the target and the proceeding point it was replaced with the value of the proceeding point.

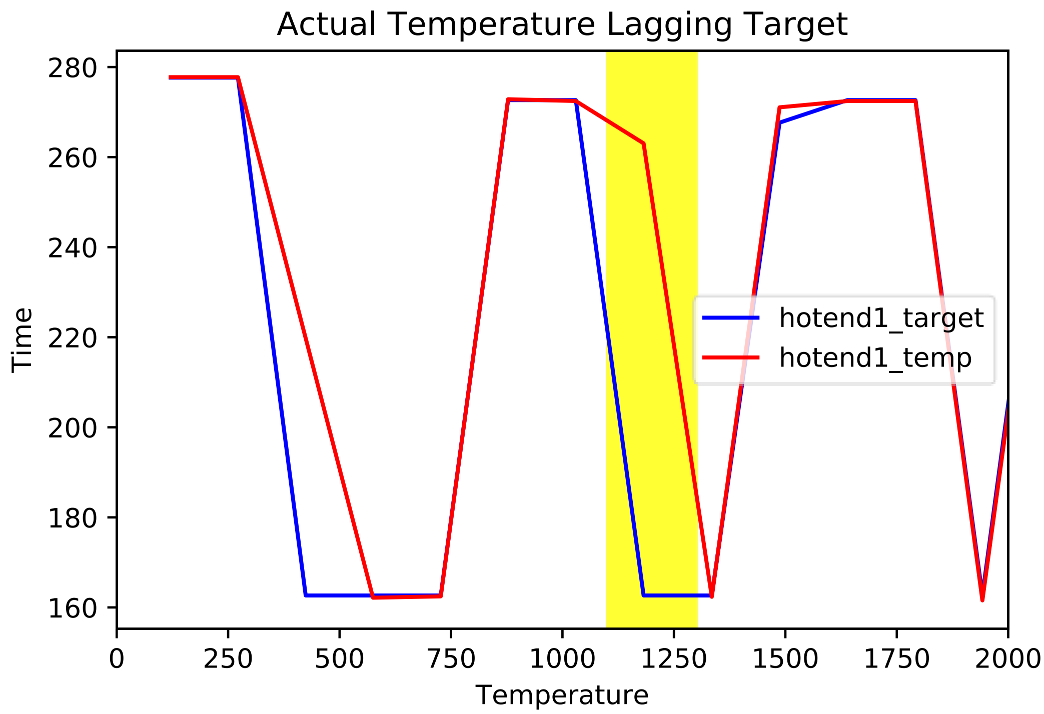

Figure 3.3 Example of large gap between temp and target caused by lag

The bed temperature and bed target have very little variability and are hard to see on this scale, even exploded in Figure 3.4 they have very little variation compared to the head temperatures and appear to have a slight upward trend. This graph is also created using Seaborn[81].

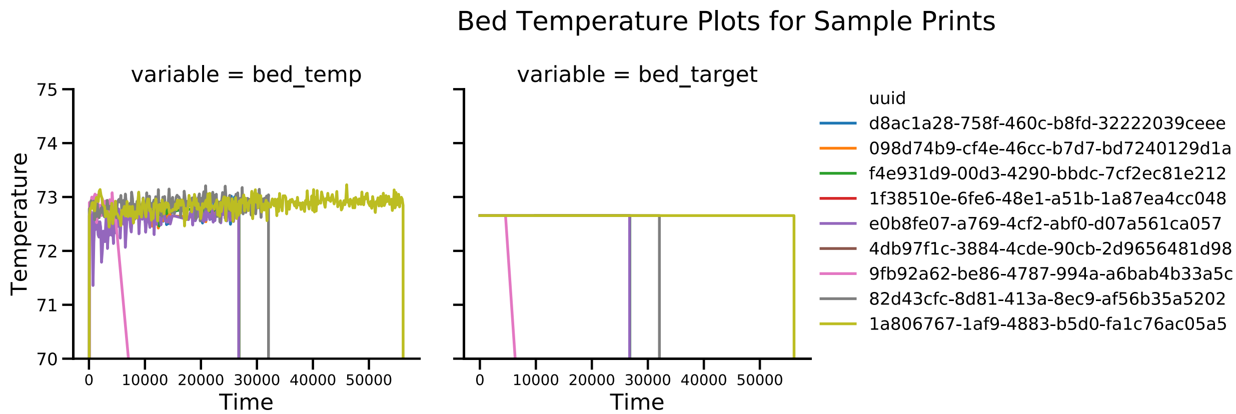

Figure 3.4 Bed temperature plot details

The Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test is a statistical test to determine if a series is stationary around a trend [83]. A stationary series among other properties has the same mean and variance regardless of when in time it was selected. The Statsmodels package in Python has an implementation that test both for stationary, and stationary around a deterministic trend [84]. Only the data from the time where the printer was printing are included to eliminate the nearly vertical lines at the beginning and end of the prints. This filtering was applied to the data for the analysis also. The KPSS test was conducted against the longest running test with the most data, with the results shown in Table 2.1. The results show that the two hotend temperatures are stationary, while the bed temperature is trend stationary. Differentiating the bed temperature by taking the delta between it and the preceding point removed the trend and made it stationary.

Table 3.3 P-Values of KPSS Test for three temperature variables

VARIABLE	P-VALUE (STATIONARY)	P-VALUE(TREND)
Hotend 1 Temperature	0. 1	0. 1
Hotend 2 Temperature	0. 1	0. 1
Bed Temperature	0. 01	0. 1
Bed Temp (delta)	0. 1	0.1

3.3 Feature Engineering

The next step in the process was feature engineering, selecting the data from the machines that could be relevant to detecting problems. The most obvious features were the temperatures from Figure 3.1. Each machine has two print heads, called hotends, and a print bed, each of which has a target and current temperature. In addition, we could create some additional features from these measurements. The difference between target and actual is one way to compare actual temperatures when the targets are different. Because of the lag in response that we saw in hotend differences between temperature as the target moved, the difference between the hotend and the previous target, and the one before that might also be useful. There were also a couple of numerical features that looked interesting but were discarded. There was a variable for fan power, but it never changed value, it was always 100%. There were also two statistical variables reported by the printer for each hotend, the maximum temperature exposed, and the material extruded. These variables behaved oddly, occasionally decreasing, the hotend max temperature exposed would even decrease when the hotend temperature increased. No explanation was found for these behaviors, so the variables were discarded. There was also one meaningful categorical variable, the material at each hotend.

CHAPTER 4: NOVELTY DETECTION MODELLING

4.1 Local Optimization Factor Model

In our survey there were no examples of using PredM in additive manufacturing, however it should be like medical device, semi-conductor manufacturing, and wind turbine industries that use modern equipment with embedded sensors and networking ability. These industries have successfully used supervised learning to develop models. One of the issues with supervised learning is the difficulty in getting labelled data, this was noted even in the aircraft industry [50] which is required to maintain records for safety purposes. The difficulty is compounded in a less regulated industry with lower margins. Other fields have addressed this challenge by using 3rd party services to crowd source the labelling of the data at a low cost[85]. Crowdsourced data has been found to be noisier that traditional in-house labelling [86], and exposes proprietary information, making it unsuitable for this work.

In addition to the cost of labelling with a process expert, the difficulty in finding examples of faults led to the use of a one-class SVM in an HVAC system solution[43]. We tried to address both issues by intentionally creating failures. Because failures are hard to predict we would need to collect data for a long time to get representative failures. After consulting with the SMEs for the system we determined that there were two failures that are seen in production that are easy to create. The first is tangling or breakage of the filament, this results in a failed part because of material not getting to a print head. The second is more serious, failure of the cooling fan. In addition to ruining the part and wasting the printing time, it can cascade to a print head failure. By simulating these failures, we can get data to analyze much faster than by waiting for one to occur. Additionally, because we caused the failure, labelling the initiation of failure becomes a trivial task.

At this point, once we had a data collection system in place, the plans to conduct an experiment were interrupted by the novel coronavirus pandemic. First, the facility was closed due to local #SaferAtHome orders. Then it re-opened to produce face shields, this urgent need eliminated the opportunity to conduct experiments, and required a change in direction. This led to a second look at an unsupervised approach. Several other examples of unsupervised learning for PredM are a comparison of multiple unsupervised models on vibration data from HVAC equipment [24], use of a prefix tree on simulated factories [25], and a cohort-based method AdaBoost [26].

The expectation was that failures we could detect with the system would be rare. After reviewing the outlier and novelty detection methods of the Python Sci-kit Learn [87] we chose to use the Local Outlier Factor method [88]. The use of novelty vs outlier depends on the training and evaluation sets of the application. For novelty detection we assume that all the runs of the training set are not anomalous, while for outlier detection it assumes that there may be outliers in the training dataset also. The LOF model is a density-based model. A point is part of a cluster if it is close to points like its neighbors are. Close is determined dynamically, by the distance from its neighbors. This allows a small dense cluster in one portion of the input space and a larger more diffuse cluster in another portion.

To determine if this algorithm is capable of detecting outliers it was tested against the Nasa Turbofan Engine Degradation set used by many of related works [45]. This data set consists of multiple settings and sensors for 100 aircraft engines from an unknown starting condition to failure. This results in 128 to 362 measurements for each engine. To match this data with the expected experimental data the cycles to failure data inherent in the dataset was not used. Instead the first 28 rows of the data set were selected as “normal” or “healthy” state. For ground truth we do not know when the first sign of degradation occurs, just that there is a failure on the last cycle. Because of the large number of anomalies, the data was debounced the data to look for 3 successive anomalous data points. The results are shown in

Figure 4.1, all the engines flagged an anomaly before failure.

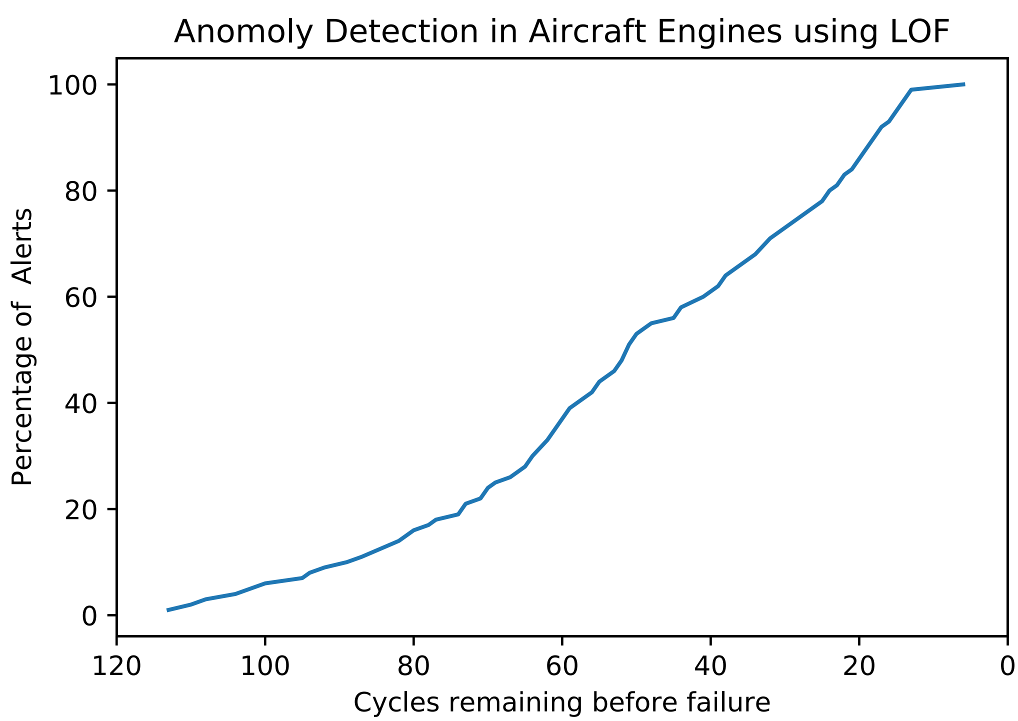

Figure 4.1 Cycles remining before failure detected on aircraft engine data with LOF

Another algorithm commonly used for novelty detection is the One-Class SVM [89]. Where a traditional binary classifier SVM uses support vectors to create a decision boundary between two classes, a one-class SVM creates a boundary around the provided class and looks for new data to be inside or outside of that boundary. It again was implemented on the same data using Sci-kit Learn. The SVM was more aggressive finding outliers and was debounced 5 times, even with that the results were still more aggressive than LOF, finding failures earlier as shown in Figure 4.2

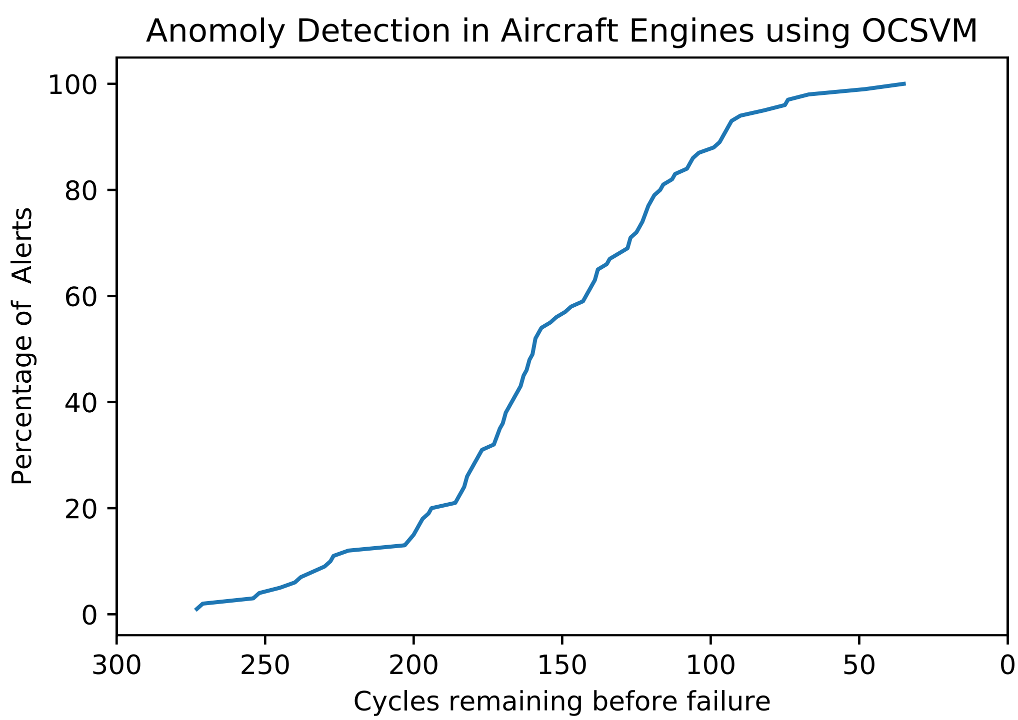

Figure 4.2 Aircraft Engine Anomoly Detection with a one-class SVM

Based on the results on from the aircraft engines set an LOF was selected for the data set. In addition to changing to Novelty detection, there were additional challenges caused by the change in the operating environments of the printers. As part of the new mode of mass-producing face shields the part had a simple geometry and only required a single material with one print head. This eliminated multiple features as shown in Table 4.1, leaving only a handful of features.

Table 4.1 Impact of Production Change on Features

FEATURE	DESCRIPTION	IMPACT OF COVID19
Hotend1_temp	Actual temperature at Extruder 1	
HOTEND1_TARGET	Desired Temperature at Extruder1	
HOTEND1_Err	Delta between Temperature and Target	
HOTEND1_LAG1	Delta between Temperature and prior Target	Redundant, the same as HOTEND1_ERR
HOTEND1_LAG2	Delta between Temperature and Target from measurement before previous measurement	Redundant, the same as HOTEND1_ERR
Hotend2_temp	Actual temperature at Extruder 2	
HOTEND2_TARGET	Desired Temperature at Extruder2	Eliminated, always 0
HOTEND2_Err	Delta between Temperature and Target	Redundant, same as HOTEND2_TARGET
HOTEND2_LAG1	Delta between Temperature and prior Target	Redundant, the same as HOTEND2_ERR
HOTEND2_LAG2	Delta between Temperature and prior Target	Redundant, the same as HOTEND2_ERR
Material1	Material being dispensed by Extruder1	Unchanging
MATERIAL2	Material being dispensed by Extruder1	Not used, no material extruded

With the system collecting data in place, and the printers making face shields data collection started in earnest. There was a limited categorization of data by the equipment itself, if the print run was completed it was classified as “Finished”, if the print run was interrupted by a machine fault it was classified as “Failed”, and if it was interrupted by the operator it was classified as “Aborted”. After the first week of data collection we had approximately 80 “Finished” runs , one “Aborted” run and no failures. The approximately 3000 rows of data associated with the “Finished” runs was extracted from the database and was used for training the model.

The model was then evaluated against another batch of approximately 2400 rows from 120 “Finished”, 3 “Failed”, and 2 “Aborted” runs, while the system continued to collect the test data set. The evaluation set was used for feature selection, as shown in Table 4.2 the model was able to distinguish the Failed runs from the Aborted and Finished runs. All the models that included ‘hotend1_temp’ and ‘hotend2_temp’ had similar performance. Where on a linear model normally the excess features would be eliminated to reduce over-fitting, on this model I kept them to increase the opportunity to detect an anomaly associated with a failure mode not seen in the evaluation set. A similar analysis on the two hyper-parameters of the LOF model, the number of nearest neighbors and the contamination, shown in Table 4.3 Impact of Hyper-Parameters on LOF Model resulted in the best F1 score using the SciKit learn recommended values of 20 and ‘auto’.

Table 4.2 Precision and Recall vs Feature Selection

Features	Precision (F)	Recall (F)	Precision (A)	Recall(A)
('hotend1_temp', 'hotend1_target', 'hotend1_err', 'hotend2_temp', 'bed_change')	. 636	. 024	. 056	. 003
('hotend1_temp', 'hotend1_target', 'hotend1_err', 'hotend2_temp')	. 636	. 024	. 056	. 003
('hotend1_temp', 'hotend1_target', 'hotend1_err')	. 091	. 033	. 019	. 011
('hotend1_temp', 'hotend1_target', 'hotend2_temp')	. 636	. 023	. 093	. 006
('hotend1_temp', 'hotend2_temp')	. 636	. 024	. 096	. 006
('hotend2_temp')	. 121	. 019	. 148	. 039

Table 4.3 Impact of Hyper-Parameters on LOF Model

Neighbors	Contamination	Precision (F)	Recall (F)	F1-Score	Precision (A)	Recall (A)
20	auto	0.636	0.024	0.046	0.056	0.003
30	auto	0.636	0.024	0.046	0.056	0.003
10	auto	0.576	0.022	0.042	0.037	0.003
20	0.2	0.727	0.021	0.040	0.017	0.008
20	0.3	0.818	0.021	0.040	0.024	0.010
20	0.1	0.666	0.022	0.042	0.074	0.004
20	0.05	0.636	0.023	0.044	0.056	0.004
20	0.01	0.212	0.018	0.034	0	0

4.2 Results

There are multiple differences between the Aircraft Engine and 3D Printer data sets. The one that made the largest impact on the result evaluation was that all 100 engines in the validation set eventually failed. There was no way to tell if the anomaly were valid or a false alarm because they all came from an engine that would fail in the provided data. There was no ground truth available on individual samples to say this was a True Positive or this was a False Negative. All that could be determined was that for both algorithms an anomaly was detected on every engine before the failure. From the data it is also apparent that the OCSVM detected the anomaly earlier and with a wider spread of detection times, but it is unknown if that is a better or worse characterization of the data set. After developing the model, it was tested on an additional data set provided as part of the Aircraft Engines Data Set. This set also consisted of 100 engines operating under the same conditions as the training set but includes two different failure modes. The results shown in Figure 4.3 are consistent with the evaluation results.

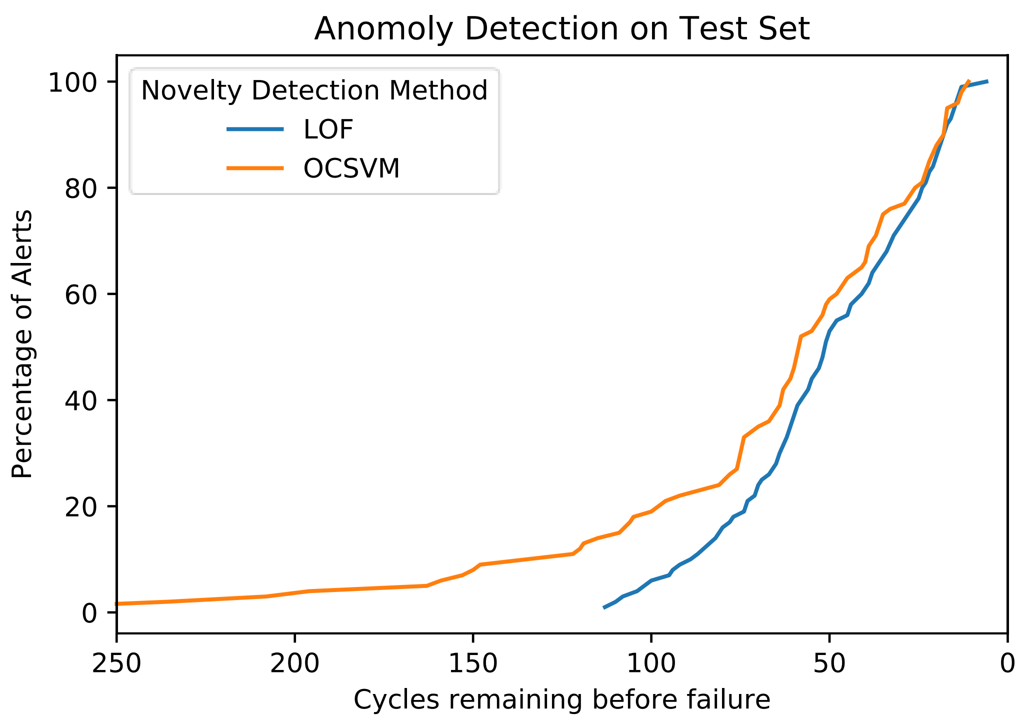

Figure 4.3 Comparison of LOF and OCSVM on Test Set

The 3D printer data was also evaluated on a new set of data, consisting of several more days of production. The results were like the test data on Failures, with a recall of 0.697 and precision of 0.050, and better on the Aborted runs with a recall of 0.512 and a precision of 0.031. Truth table is shown in Table 4.4, note that positive results (anomalies) are shown as -1 as returned by the LOF model.

Table 4.4 Truth Table for 3D Printer Results

ACTUAL PREDICTED	ABORTED	FAILED	FINISHED
-1	29	46	840
1	27	20	1585

Because the test data was labeled, we know what type of failures were detected and how many anomalous pts were detected in each. These details are shown in Table 4.5. The filament failure which was planned for the original experiment has a relatively weak signal as compared to the Comm Error. There were also some heater failures included in the mix.

Table 4.5 Details of Failures Detected

UUID	RESULT	FAILURE MODE DETAIL	ANOM PTS	NORMAL PTS
5c6b	Failed	Comm Err	5	1
df1d	Failed	Comm Err	15	1
edd0	Failed	Comm Err	3	1
0f42	Failed	Heater Failure	6	10
d508	Failed	Heater Failure	8	1
1473	Abort	Filament	16	21
dea6	Failed	Heater Failure	9	6
2329	Abort	Filament	13	6

The results of any individual data point do not tell the whole story, there is also the option to look at the whole cycle. To do this we calculated the percentage of failures in any given cycle. Table 4.6 shows the truth table with a cutoff selected to maximize the F1 Score of the Failure class. It is obvious that the recall has improved to 100% on both types of faults. For the failures, the precision improved from 0.050 to 0.171, and on the Aborted class it improved from 0.031 to 0.057. This improved the F1 on the Failures from 0.09 to 0.29 and on the Aborted from 0.058 to 0.11. The tradeoff is that we end up with significantly less data.

Table 4.6 Truth Table for complete cycles

Actual Predicted	Aborted	Failed	Finished
-1	2	6	27
1	0	0	52

Varying the cutoff allows the calculation the precision recall curve. A precision-recall curve is the preferred method for validating the effectiveness of an imbalanced dataset because the ROC curve may provide an optimistic view of performance[90]. To make the P-R Curve Figure 4.4 we combined the 'Aborted' and 'Failed' results, as they both indicate an unsuccessful print. We can see that it has better performance that a random model.

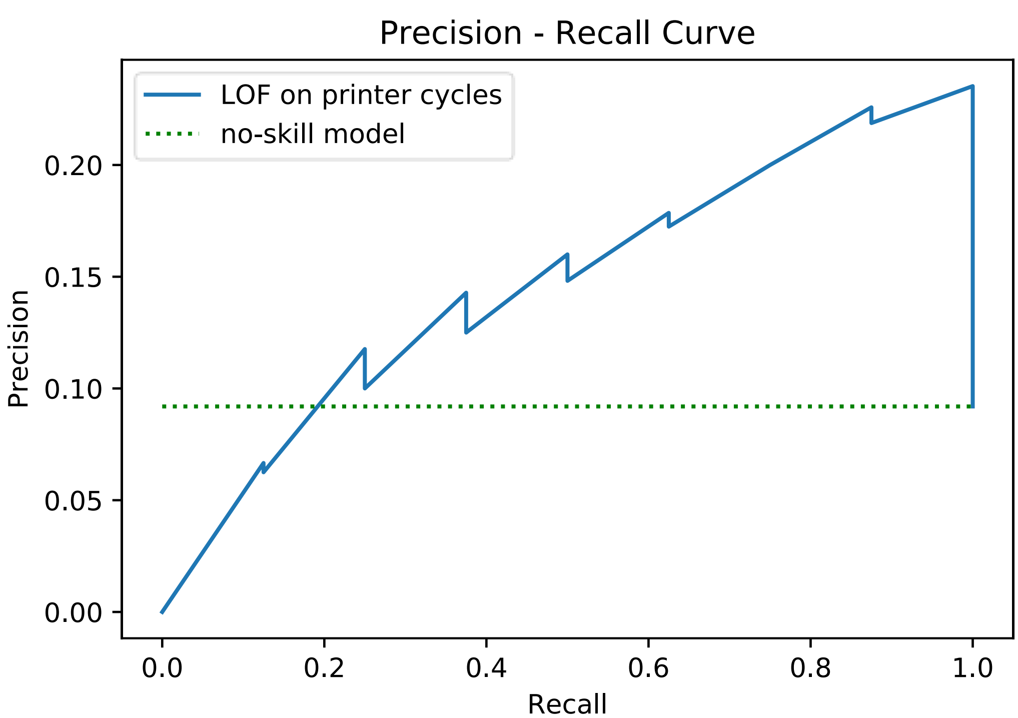

Figure 4.4 Precision Recall Curve of the LOF model evaluating complete cycles of the printer..

From this curve we see that the anomaly model returned several "normal" points first causing the P-R curve to start at the lower left and then grow as actual "novel" points are returned. Ideally, we would like the plot to start out at the top left, retuning "novel points first. By comparison with the green dotted line we can see the boost over a random model.

One way to improve the model would be to add additional data, however after building all the face shields needed by the local hospitals the site returned to #SaferAtHome mode and ceased production. To test the effects of more data, the evaluation set was cannibalized and all the points that were not failures and were more than 5 runs before a failure from the eval set were included in the training set. The model was then retrained using the same features and hyperparameters as the first run. This provided better results, with the Precision-Recall curve shown in Figure 4.5. The need for more data is discussed in the conclusion.

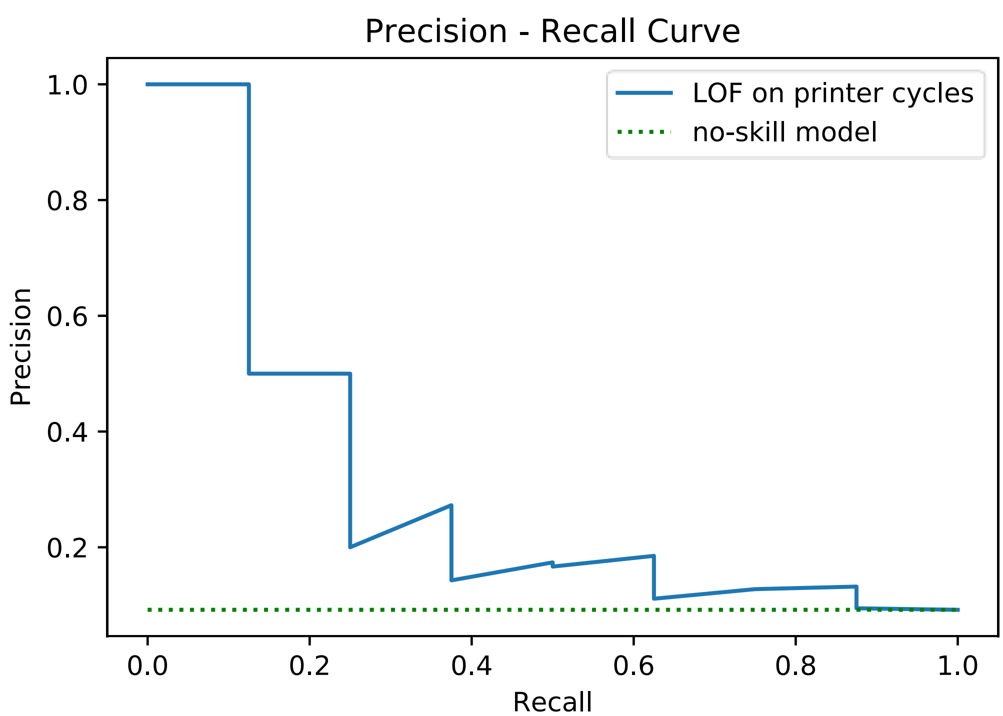

Figure 4.5 P-R curve after retraining on more data. Same features, hyperparameters, and test set as previous.

After reviewing the results, the first 3 points returned are the “Comm Error” failures. The model does a good job of detecting these but is not able to detect “Heater” and “Filament” faults.

CHAPTER 5: CONCLUSION

This work demonstrated that novelty detection could be used successfully to detect failures in both a public data set and a real-world set collect for this purpose. The advantages of the anomaly detection were that it took very little data to test and was capable of detecting failure modes that it had not seen in training. This was offset by the reduced precision, either in the wide range of predicted failure warning on the Aircraft Engines set or the high false positive rate on the 3D Printer data set. This can be seen in that the Aircraft Engine model was trained on only 2500 rows of the 20,000 in the data set. A typical classification model would have left 30% for validation, meaning 14,000 rows for training, over 5 times the data. The additive data was trained on only 3000 rows of data originally, and the retrained model still only has about 5200.

There are several areas for investigation to improve these results using anomaly detection. The first is to take advantage of the Unreasonable Effectiveness of Data [55] and continue to gather more data. Increasing the data available has been shown to help machine learning models. This can be accomplished just with time, as the machine continues to make parts more data will be collected. It can be accelerated by increasing the rate of data collection from several minutes between points to collecting data every 10-20 seconds. A second option is to investigate additional anomaly detection algorithms that have been used in the literature, both dissimilarity-based [44] and neural network based [18].

The original reason for selecting an anomaly-based method was the difficulty and cost in obtaining and labelling the failure data. An alternate route would be to use a classification model, and instead of investigating a better algorithm, spend the effort on a Software Engineering solution to label data more easily. Labelling the test samples was a manual process involving email. An additional advantage of anomaly detection was not needing a large number of failures to create a model, but if the anomaly detection model needs a significant amount more training data, then the additional time to collect normal samples will also collect additional anomalous samples.

This work also treated the points independently, ignoring the time series nature of the data. Possibly that anomalies are occurring multiple cycles before failure are not false positives, further investigation of this possibility could have significant economic benefits. It is also possible that the random spikes in Figure 3.2 are signal and not noise.

This work has demonstrated the ability to use novelty detection for predictive maintenance. Based on the results achieved it is suitable for a situation where faults are rare, and the cost of false positives is low.

REFERENCES

[1] "Innovation / Leben 4.0 / Bundesverband der Deutschen Industrie e.V. ," *Leben 4.0 / Bundesverband der Deutschen Industrie e.V.* [Online]. Available: http://www.bdi.eu/leben-4.0/innovation. [Accessed: 12-Feb-2020].

[2] "StackPath. " [Online]. Available: https://www.machinedesign.com/automation-iiot/article/21834312/the-internet-of-industrial-things. [Accessed: 12-Feb-2020].

[3] Amazon Web Services, "Top Use Cases for Industrial IoT Applications. " [Online]. Available: https://pages.awscloud.com/industrial-iot-top-use-cases-ebook.html. [Accessed: 11-Feb-2020].

[4] J. H. Williams, A. Davies, and P. R. Drake, Eds. , *Condition-based maintenance and machine diagnostics*, First edition. London Glasgow Weinheim New York: Chapman & Hall, 1994.

[5] "Predictive Maintenance 4.0 Beyond the hype: PdM 4.0 delivers results." [Online]. Available: https://dataanalytics.report/whitepapers/predictive-maintenance-40-beyond-the-hype-pdm-40-delivers-results/5777. [Accessed: 10-Jan-2020].

[6] PTC Corporation, *The State of the Industrial Internet of Things 2017*. [Online]. Available: https://www.ptc.com/-/media/Files/PDFs/IoT/State-of-IIoT-Whitepaper.pdf. [Accessed: 11-Feb-2020].

[7] D. Dimitrov, K. Schreve, and B. N. de, "Advances in three dimensional printing – state of the art and future perspectives," *Rapid Prototyping Journal*, vol. 12, no. 3, pp. 136–147, Jan. 2006, doi: 10.1108/13552540610670717.

[8] M. F. Finch and N. Tabrizi, "Machine Learning for Predictive Maintenance: A Systemic Review of Literature" *IEEE Transactions on Knowledge and Data Engineering*, Submitted 5/29/2020

[9] P. Strauß, M. Schmitz, R. Wöstmann, and J. Deuse, "Enabling of Predictive Maintenance in the Brownfield through Low-Cost Sensors, an IIoT-Architecture and Machine Learning," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 1474–1483, doi: 10.1109/BigData.2018.8622076.

[10] J. Zheng and A. Dagnino, "An initial study of predictive machine learning analytics on large volumes of historical data for power system applications," in *2014 IEEE International Conference on Big Data (Big Data)*, Oct. 2014, pp. 952–959, doi: 10.1109/BigData.2014.7004327.

[11] D. Wu, C. Jennings, J. Terpenny, and S. Kumara, "Cloud-based machine learning for predictive analytics: Tool wear prediction in milling," in *2016 IEEE International Conference on Big Data (Big Data)*, Dec. 2016, pp. 2062–2069, doi: 10.1109/BigData.2016.7840831.

[12] B. Kroll, D. Schaffranek, S. Schriegel, and O. Niggemann, "System modeling based on machine learning for anomaly detection and predictive maintenance in industrial plants," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014, pp. 1–7, doi: 10.1109/ETFA.2014.7005202.

[13] G. A. Susto and A. Beghi, "Dealing with time-series data in Predictive Maintenance problems," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–4, doi: 10.1109/ETFA.2016.7733659.

[14] B. Luo, H. Wang, H. Liu, B. Li, and F. Peng, "Early Fault Detection of Machine Tools Based on Deep Learning and Dynamic Identification," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 509–518, Jan. 2019, doi: 10.1109/TIE.2018.2807414.

[15] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1539–1548, Feb. 2018, doi: 10.1109/TIE.2017.2733438.

[16] R. Pinto and T. Cerquitelli, "Robot fault detection and remaining life estimation for predictive maintenance," *Procedia Computer Science*, vol. 151, pp. 709–716, Jan. 2019, doi: 10.1016/j.procs.2019.04.094.

[17] E. Fumeo, L. Oneto, and D. Anguita, "Condition Based Maintenance in Railway Transportation Systems Based on Big Data Streaming Analysis," Procedia Computer Science, vol. 53, pp. 437–446, Jan. 2015, doi: 10.1016/j.procs.2015.07.321.

[18] P. K. Gopalakrishnan, B. Kar, S. K. Bose, M. Roy, and A. Basu, "Live Demonstration: Autoencoder-Based Predictive Maintenance for IoT," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–1, doi: 10.1109/ISCAS.2019.8702230.

[19] K. Mishra and S. K. Manjhi, "Failure Prediction Model for Predictive Maintenance," in *2018 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 2018, pp. 72–75, doi: 10.1109/CCEM.2018.00019.

[20] P. Pirasteh *et al.*, "Interactive feature extraction for diagnostic trouble codes in predictive maintenance: A case study from automotive domain," in *Proceedings of the Workshop on Interactive Data Mining*, Melbourne, VIC, Australia, Feb. 2019, pp. 1–10, doi:10.1145/3304079.3310288.

[21] R. Prytz, S. Nowaczyk, T. Rögnvaldsson, and S. Byttner, "Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data," *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 139–150, May 2015, doi: 10.1016/j.engappai.2015.02.009.

[22] H. Li *et al.* , "Improving rail network velocity: A machine learning approach to predictive maintenance," *Transportation Research Part C: Emerging Technologies*, vol. 45, pp. 17–26, Aug. 2014, doi: 10.1016/j.trc.2014.04.013.

[23] D. Apiletti *et al.* , "iSTEP, an Integrated Self-Tuning Engine for Predictive Maintenance in Industry 4. 0," in *2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, 2018, pp. 924–931, doi: 10.1109/BDCloud.2018.00136.

[24] "Backblaze Hard Drive Stats. " [Online]. Available: https://www.backblaze.com/b2/hard-drive-test-data.html. [Accessed: 25-Feb-2020].

[25] S. Xiang, D. Huang, and X. Li, "A Generalized Predictive Framework for Data Driven Prognostics and Diagnostics using Machine Logs," in *TENCON 2018 - 2018 IEEE Region 10 Conference*, 2018, pp. 0695–0700, doi: 10.1109/TENCON.2018.8650152.

[26] "Hortonworks: Analyze HVAC Machine and Sensor Data (2017)." [Online] Available: http://s3.amazonaws.com/hw-sandbox/tutorial14/SensorFiles.zip [Accessed: 25-Feb-2020]

[27] I. S. Candanedo, E. H. Nieves, S. R. González, M. T. S. Martín, and A. G. Briones, "Machine Learning Predictive Model for Industry 4. 0," in *Knowledge Management in Organizations*, Cham, 2018, pp. 501–510, doi: 10.1007/978-3-319-95204-8_42.

[28] R. Sipos, D. Fradkin, F. Moerchen and Z. Wang "Log-based predictive maintenance | Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. " doi: 10.1145/2623330.2623340.

[29] X. Li, "Remaining Useful Life Prediction of Bearings Using Fuzzy Multimodal Extreme Learning Regression," in *2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, 2017, pp. 499–503, doi: 10.1109/SDPC.2017.100.

[30] P. Qian, X. Ma, and Y. Wang, "Condition monitoring of wind turbines based on extreme learning machine," in *2015 21st International Conference on Automation and Computing (ICAC)*, 2015, pp. 1–6, doi: 10.1109/IConAC.2015.7313974.

[31] S. K. Bose, B. Kar, M. Roy, P. K. Gopalakrishnan, and A. Basu, "ADEPOS: anomaly detection based power saving for predictive maintenance using edge computing," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference on - ASPDAC '19*, Tokyo, Japan, 2019, pp. 597–602, doi: 10.1145/3287624.3287716.

[32] L. Liao and H. Ahn, "Combining Deep Learning and Survival Analysis for Asset Health Management" | PHM Society. [Online]. Available: https://www.phmsociety.org/node/2192. [Accessed: 24-Feb-2020].

[33] T. Huuhtanen and A. Jung, "PREDICTIVE MAINTENANCE OF PHOTOVOLTAIC PANELS VIA DEEP LEARNING," in *2018 IEEE Data Science Workshop (DSW)*, 2018, pp. 66–70, doi: 10.1109/DSW.2018.8439898.

[34] D. Wu, C. Jennings, J. Terpenny, R. X. Gao, and S. Kumara, “A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests,” *J. Manuf. Sci. Eng*, vol. 139, no. 7, Jul. 2017, doi: 10.1115/1.4036350.

[35] A. Kanawaday and A. Sane, “Machine learning for predictive maintenance of industrial machines using IoT sensor data,” in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2017, pp. 87–90, doi: 10.1109/ICSESS.2017.8342870.

[36] L. Salmaso *et al.* , “Design of experiments and machine learning to improve robustness of predictive maintenance with application to a real case study,” *Communications in Statistics - Simulation and Computation*, vol. 0, no. 0, pp. 1–13, Sep. 2019, doi: 10.1080/03610918.2019.1656740.

[37] K. Liulys, “Machine Learning Application in Predictive Maintenance,” in *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 2019, pp. 1–4, doi: 10.1109/eStream.2019.8732146.

[38] S. Butte, P. A R, and S. Patil, “Machine Learning Based Predictive Maintenance Strategy: A Super Learning Approach with Deep Neural Networks,” in *2018 IEEE Workshop on Microelectronics and Electron Devices (WMED)*, 2018, pp. 1–5, doi: 10.1109/WMED.2018.8360836.

[39] C. H. Yeh, M. H. Lin, C. H. Lin, C. E. Yu, and M. J. Chen, “Machine Learning for Long Cycle Maintenance Prediction of Wind Turbine,” *Sensors*, vol. 19, no. 7, p. 1671, Jan. 2019, doi: 10.3390/s19071671.

[40] M. Baptista, S. Sankararaman, Ivo. P. de Medeiros, C. Nascimento, H. Prendinger, and E. M. P. Henriques, “Forecasting fault events for predictive maintenance using data-driven techniques and ARMA modeling,” *Computers & Industrial Engineering*, vol. 115, pp. 41–53, Jan. 2018, doi: 10.1016/j.cie.2017.10.033.

[41] F. Francis and M. Mohan, “ARIMA Model based Real Time Trend Analysis for Predictive Maintenance,” in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2019, pp. 735–739, doi: 10.1109/ICECA. 2019. 8822191.

[42] N. Amruthnath and T. Gupta, “A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance,” in *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, 2018, pp. 355–361, doi: 10.1109/IEA. 2018. 8387124.

[43] A. Beghi, L. Cecchinato, C. Corazzol, M. Rampazzo, F. Simmini, and G. A. Susto, “A One-Class SVM Based Tool for Machine Learning Novelty Detection in HVAC Chiller Systems,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1953–1958, Jan. 2014, doi: 10.3182/20140824-6-ZA-1003.02382.

[44] R. Satta, S. Cavallari, E. Pomponi, D. Grasselli, D. Picheo, and C. Annis, "A dissimilarity-based approach to predictive maintenance with application to HVAC systems," *arXiv:1701.03633 [csx]*, Jan. 2017.

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

[46] V. Mathew, T. Toby, V. Singh, B. M. Rao, and M. G. Kumar, "Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning," in *2017 IEEE International Conference on Circuits and Systems (ICCS)*, 2017, pp. 306–311, doi: 10.1109/ICCS1.2017.8326010.

[47] D. Bruneo and F. De Vita, "On the Use of LSTM Networks for Predictive Maintenance in Smart Industries," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2019, pp. 241–248, doi: 10.1109/SMARTCOMP.2019.00059.

[48] A. El Afia and M. Sarhani, "Particle Swarm Optimization for Model Selection of Aircraft Maintenance Predictive Models," in *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*, Tetouan, Morocco, Mar. 2017, pp. 1–12, doi: 10.1145/3090354.3090402.

[49] G. Kejela, R. M. Esteves, and C. Rong, "Predictive Analytics of Sensor Data Using Distributed Machine Learning Techniques," in *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, 2014, pp. 626–631, doi: 10.1109/CloudCom.2014.44.

[50] C. Yang, S. Letourneau, J. Liu, Q. Cheng, and Y. Yang, "Machine learning-based methods for TTF estimation with application to APU prognostics," *Appl Intell*, vol. 46, no. 1, pp. 227–239, Jan. 2017, doi: 10.1007/s10489-016-0829-4

[51] G. A. Susto et al., "An adaptive machine learning decision system for flexible predictive maintenance," in 2014 IEEE International Conference on Automation Science and Engineering (CASE), 2014, pp. 806–811, doi: 10.1109/CoASE.2014.6899418.

[52] R. B. Patil, M. A. Patil, V. Ravi, and S. Naik, "Predictive modeling for corrective maintenance of imaging devices from machine logs," in 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017, pp. 1676–1679, doi: 10.1109/EMBC.2017.8037163.

[53] A. Kumar *et al.*, "Using Machine Learning to Assess the Risk of and Prevent Water Main Breaks," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, United Kingdom, Jul. 2018, pp. 472–480, doi: 10.1145/3219819.3219835.

[54] A. Binding, N. Dykeman, and S. Pang, “Machine Learning Predictive Maintenance on Data in the Wild,” in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, pp. 507–512, doi: 10.1109/WF-IoT.2019.8767312.

[55] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, pp. 843–852, doi: 10.1109/ICCV.2017.97.

[56] B. Cline, R. S. Niculescu, D. Huffman, and B. Deckel, “Predictive maintenance applications for machine learning,” in *2017 Annual Reliability and Maintainability Symposium (RAMS)*, 2017, pp. 1–7, doi: 10.1109/RAM.2017.7889679.

[57] S. Xayyasith, A. Promwungkwa[, and K. Ngamsanroaj, “Application of Machine Learning for Predictive Maintenance Cooling System in Nam Ngum-1 Hydropower Plant,” in *2018 16th International Conference on ICT and Knowledge Engineering (ICT KE)*, 2018, pp. 1–5, doi: 10.1109/ICTKE.2018.8612435.

[58] T. Praveenkumar, M. Saimurugan, P. Krishnakumar, and K. I. Ramachandran, “Fault Diagnosis of Automobile Gearbox Based on Machine Learning Techniques,” *Procedia Engineering*, vol. 97, pp. 2092–2098, Jan. 2014, doi: 10.1016/j.proeng.2014.12.452.

[59] D. Kateris, D. Moshou, X. -E. Pantazi, I. Gravalos, N. Sawalhi, and S. Loutridis, “A machine learning approach for the condition monitoring of rotating machinery,” *J Mech Sci Technol*, vol. 28, no. 1, pp. 61–71, Jan. 2014, doi: 10.1007/s12206-013-1102-y.

[60] A. Coraddu, L. Oneto, A. Ghio, S. Savio, M. Figari, and D. Anguita, “Machine learning for wear forecasting of naval assets for condition-based maintenance applications,” in *2015 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles (ESARS)*, 2015, pp. 1–5, doi: 10.1109/ESARS.2015.7101499.

[61] A. Kumar, R. Shankar, and L. S. Thakur, “A big data driven sustainable manufacturing framework for condition-based maintenance prediction,” *Journal of Computational Science*, vol. 27, pp. 428–439, Jul. 2018, doi: 10.1016/j.jocs.2017.06.006.

[62] J. Lee, H. Qiu, G. Yu, J. Lin, and Rexnord Technical Services (2007). IMS, University of Cincinnati. "Bearing Data Set", NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA

[63] H. Tam, K. Lee, S. Liu, L. Cho, and K. Cheng, “Intelligent Optical Fibre Sensing Networks Facilitate Shift to Predictive Maintenance in Railway Systems,” in *2018 International Conference on Intelligent Rail Transportation (ICIRT)*, 2018, pp. 1–4, doi: 10.1109/ICIRT.2018.8641602.

[64] J. Yan, Y. Meng, L. Lu, and L. Li, “Industrial Big Data in an Industry 4. 0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance,” *IEEE Access*, vol. 5, pp. 23484–23491, 2017, doi: 10.1109/ACCESS.2017.2765544.

[65] Z. Li, Y. Wang, and K. -S. Wang, “Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4. 0 scenario,” *Adv. Manuf.* , vol. 5, no. 4, pp. 377–387, Dec. 2017, doi: 10.1007/s40436-017-0203-8.

[66] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. G. Jun, and J. W. Sutherland, “Predictive Maintenance of Machine Tool Systems Using Artificial Intelligence Techniques Applied to Machine Condition Data,” *Procedia CIRP*, vol. 80, pp. 506–511, Jan. 2019, doi: 10.1016/j.procir.2018.12.019.

[67] E. Sezer, D. Romero, F. Guedea, M. Macchi, and C. Emmanouilidis, “An Industry 4. 0-Enabled Low Cost Predictive Maintenance Approach for SMEs,” in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2018, pp. 1–8, doi: 10.1109/ICE.2018.8436307.

[68] M. Paolanti, L. Romeo, A. Felicetti, A. Mancini, E. Frontoni, and J. Loncarski, “Machine Learning approach for Predictive Maintenance in Industry 4. 0,” in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2018, pp. 1–6, doi: 10.1109/MESA.2018.8449150.

[69] S Munirathinam and B Ramadoss, “Predictive Models for Equipment Fault Detection in the Semiconductor Manufacturing Process,” in *IACSIT International Journal of Engineering and Technology, Vol. 8, No. 4, August 2016. DOI: 10.7763/IJET.2016.V8.898*

[70] N. Kolokas, T. Vafeiadis, D. Ioannidis, and D. Tzovaras, “Forecasting faults of industrial equipment using machine learning classifiers,” in *2018 Innovations in Intelligent Systems and Applications (INISTA)*, 2018, pp. 1–6, doi: 10.1109/INISTA.2018.8466309.

[71] G. A. [, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, “Machine Learning for Predictive Maintenance: A Multiple Classifier Approach,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 812–820, Jun. 2015, doi: 10.1109/TII.2014.2349359.

[72] G. K. Durbhaka and B. Selvaraj, “Predictive maintenance for wind turbine diagnostics using vibration signal analysis based on collaborative recommendation approach,” in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016, pp. 1839–1842, doi: 10.1109/ICACCI.2016.7732316.

[73] L. Colone, N. Dimitrov, and D. Strauß, “Predictive repair scheduling of wind turbine drive-train components based on machine learning,” *Wind Energy*, vol. 22, no. 9, pp. 1230–1242, 2019, doi: 10.1002/we.2352.

[74] M. Canizo, E. Onieva, A. Conde, S. Charramendieta, and S. Trujillo, “Real-time predictive maintenance for wind turbines using Big Data frameworks,” in *2017 IEEE International*

Conference on Prognostics and Health Management (ICPHM), 2017, pp. 70–77, doi: 10.1109/ICPHM.2017.7998308.

[75] F. Kabir, B. Foggo, and N. Yu, "Data Driven Predictive Maintenance of Distribution Transformers," in *2018 China International Conference on Electricity Distribution (CICED)*, 2018, pp. 312–316, doi: 10.1109/CICED.2018.8592417.

[76] J. Grambau, A. Hitzges, and B. Otto, "Reference Architecture framework for enhanced social media data analytics for Predictive Maintenance models," in *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2019, pp. 1–8, doi: 10.1109/ICE.2019.8792678.

[77] W. Gao *et al.*, "The status, challenges, and future of additive manufacturing in engineering," *Computer-Aided Design*, vol. 69, pp. 65–89, Dec. 2015, doi: 10.1016/j.cad.2015.04.001.

[78] "Ultimaker S3: Massive performance, impressively small footprint," *ultimaker.com*. [Online]. Available: https://ultimaker.com/3d-printers/ultimaker-s3. [Accessed: 06-Mar-2020].

[79] Oliphant, T.E. A guide to Numpy. Vol. 1 Trelgol Publishing USA, 2006

[80] The pandas development team, *pandas-dev/pandas: v0.23.4* Zenodo, 2018.

[81] Michael Waskom, Olga Botvinnik, Drew O'Kane, Paul Hobson, Joel Ostblom, Saulius Lukauskas, David C Gemperline, Tom Augspurger, Yaroslav Halchenko, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Cameron Pye, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Pete Bachant, Marcel Martin, Kyle Meyer, Alistair Miles, Yoav Ram, Thomas Brunner, Tal Yarkoni, Mike Lee Williams, Constantine Evans, Clark Fitzgerald, Brian, and Adel Qalieh, "mwaskom/seaborn: v0.9.0 (July 2018)". Zenodo, 16-Jul-2018.

[82] Hunter, J.D. "Matplotlib: A 2D graphics environment." *Computing in Science & Engineering* Vol 9, p.90-95 doi: 10.5281/zenodo.1420605 2018

[83] D. Kwiatkowski , P. Phillips , P. Schmidt , and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root ," *Journal of Econometrics* , vol. 54, p. 159-178, 1992.

[84] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, 2010.

[85] "Amazon SageMaker Ground Truth | AWS," *Amazon Web Services, Inc.* https://aws.amazon.com/sagemaker/groundtruth/ (accessed Apr. 07, 2020).

[86] M. Lease, "On Quality Control and Machine Learning in Crowdsourcing," in *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[87] F. Pedregosa *et al.* , “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[88] M. M. Breunig, H. -P. Kriegel, R. T. Ng, and J. Sanders, “LOF: Identifying Density-Based Local Outliers,” in *Proceedings of ACM SIGMOD*, Dallas, Tx, 2000.

[89] B. Scholkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the Support of a High-Dimensional Distribution,” *Neural Computation*, vol. 13, no. 7, Jul. 2001, doi: 10.1162/089976601750264965.

[90] P. Branco, L. Torgo, and R. P. Ribeiro, “A Survey of Predictive Modeling on Imbalanced Domains.” ACM Comput. Surv., vol.49, no. 2, pp 1-50, Nov 2016 doi: 10.1145/2907070.