Predictive Analytics for Maintenance Activities in Nuclear Power Plants: A Feasibility Study

by

Rajinder Khurmi

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Science in Computer Science

The Faculty of Science
University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

December 2021

THESIS EXAMINATION INFORMATION

Submitted by: **Rajinder Khurmi**

Masters of Science in **Computer Science**

Thesis Title: Predictive Analytics for Maintenance Activities in Nuclear Power Plants: A Feasibility Study

An oral defense of this thesis took place on December 2, 2021 in front of the following examining committee:

Examining Committee

Chair of Examining Committee	Dr. Peter Lewis
Research Supervisors	Dr. Karthik Sankaranarayanan and
Research Co-supervisor	Dr. Glenn Harvel
Examining Committee Member	Dr. Khalil El-Khatib
Thesis Examiner	Dr. Patrick Hung

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

Nuclear power plants are known for their use of legacy systems and processes. As plants age, the amount of maintenance increases while resources remain finite, leading to unwanted delays, affecting the health of assets and increasing costs. To aid in the modernization and digitization of nuclear power plants, this work explores data driven methods, including statistical and machine learning techniques to predict target variables. Representative Naval Propulsion Plant data with variables similar to that in the nuclear industry are used as nuclear data is not available in the public domain. Experimental results confirm target variables can be predicted with relatively high accuracy, with Deep Learning methods harbouring the lowest relative error. Two frameworks are developed based on results to showcase how predictive analytics can be used in nuclear power plant maintenance. This work is a proof of concept informing stakeholders that data driven approaches are viable in reducing maintenance delays.

Keywords: Nuclear Power Plants; Machine Leaning; Predictive Analytics; Digitization and Modernization; Engineering Management

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

______________________________ RAJINDER KHURMI

STATEMENT OF CONTRIBUTIONS

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

Dedicated to the lost kids

ACKNOWLEDGEMENTS

The following are the individuals and groups acknowledged:

- Dr.Sankaranarayanan and Dr.Harvel: For their innumerable amounts of knowledge, experience, patience, understanding and honesty.
- Nicholas (Not Nick) Somer: For being a great colleague, verbal sparring partner and my favourite natural contrarian.
- Kira Hawrysh: For sharing the ups and down of grad school and being one of most driven and optimistic friends I have.
- Vajran Savendran and Yvonne Lin: For being great friends and lab partners.
- My sister: For helping me keep my sanity and being there always.
- My Parents: For all the love, support and stress.
- Clay, Chris and Brian: For always being there for me and supporting what I do and checking me when I am in the wrong.
- The Fishbowl: This would not have been a possibility if it was not for all of you.

"Know what you need to know and do what you need to do" - unknown

"Get out of your rabbit holes" - Anonymous

Love Love - Anonymous

"Just remember, all caps when you spell the man name. - MF DOOM

Contents

List of Figures

List of Tables

1 Introduction

The nuclear industry is known for its use of legacy systems and processes due to stability and safety requirements. Due to the nature of the industry and the risks associated with producing electricity using nuclear energy, the industry is rightfully rooted in a culture of using deterministic approaches where possible and having high levels of conservatism in engineering and decision making.

This approach however limits the amount of advancements that have taken place in the industry, especially in analytics, when compared to other industrial settings.

However, there is a push towards modernization and digitization in the nuclear industry to leverage advanced technologies and analytics to be more competitive in the energy market [5].

One of these technologies include data-driven analytics such as machine learning in order to extract meaning, predictions and aid in decision making using existing data.

One of the key aspects in which these analytical approaches can be applied in is with Maintenance & Operations (M&O). Maintenance costs are some of the highest costs in running a nuclear powerplant and with continual aging in components and systems, maintenance costs get higher year over year [6].

An area of opportunity to improve the maintenance operations and reduce costs is by identifying and removing non-value added tasks and other cost contributors. One of these cost contributors is maintenance delays and backlog which results in a reduction of useful resource utilization as well as incurs damages to assets.

Assuming that an average skilled worker in a nuclear power plant works 40 hours a week every year at a salary of $80,000 USD, the hourly expense per worker is approximately $38 [6]. As derived by Rothwell , two 1117 MW reactors can be estimated to have a total work force of 1,356 workers [6]. Reducing delays by an hour everyday and replacing it with productive work, could potentially save upwards of approximately $13Million per year based on a 261 working day schedule.

As such, the motivation for this work is to explore the problem of maintenance delays using state of the art data-driven approaches. Also, it is to be explored on how to implement these approaches in nuclear power plants and other settings to aid decision makers and maintenance planners.

1.1 Problem Statement

As nuclear power plants age, the amount of maintenance required to keep them operational increases. Most maintenance tasks within power plants are still periodic and follow set schedules, however, with resource limitations and availability requirements, the amount of maintenance that can occur at any given moment is limited. This limitation causes tasks to be delayed and as a result, many maintenance tasks are deferred into maintenance backlog which is undesirable from a reliability, safety and financial perspective. Thus, a way to predict maintenance delays is needed to aid in the planning and reduction of backlog maintenance.

There are 2 aspects that this thesis covers.

1. Investigate planning and operational configurations which result in delayed tasks causing backlog maintenance.

2. Investigate various data-driven methods to be able to predict maintenance delays and how predictions can aid in reducing maintenance backlog.

1.2 Objectives

The objective of this thesis are as followed:

1. Investigate how maintenance is planned in nuclear power plants.

2. Identify nuclear power plant-specific maintenance planning considerations.

3. Correlate maintenance delays to deferred maintenance and their consequences.

4. Explore and test data-driven methods to predict a target variable related to maintenance task delays.

5. Develop a proof of concept framework(s) based on the results of the tested data-driven methods to better understand how to implement and work with data-driven analytics for maintenance activities.

1.2.1 Application of Objectives to Problem Statement

Objectives 1, 2 & 3 are to aid in the understanding of the mechanisms of maintenance operations and planning that may result in maintenance delays and subsequently deferrals. These objectives also aid in understanding of the specific considerations in nuclear power plants and how they differ from a generalized approach.

Objective 4 & 5 help develop an understanding of various types of data-driven approaches and how to implement them.

1.3 Thesis Structure

The thesis structure is as follows:

- Chapter 2 explores the literature and practices behind maintenance planning and operations in nuclear power plants.
- Chapter 3 explores the literature behind different data driven analytical approaches with a focus on supervised learning.
- Chapter 4 describes the methods and approaches used to aid in the development of the frameworks.
- Chapter 5 presents the results of the various tests with the approaches described in chapter 4.
- Chapter 6 presents the frameworks derived from the information in chapter 5 and comments on how the frameworks are to be implemented and how they aid in reducing maintenance delays and deferrals
- Chapter 7 summaries and concludes this thesis.

2 Literature Review

2.1 Maintenance Practices In Industry

Maintenance in industrial applications is the process of inspecting, verifying performance, servicing, repair and replacement of assets in a process system. The benefit of maintenance is to allow for prolonged operation of an asset at safer, efficient and more cost effective levels than operating until asset failure.

All industrial applications have some sort of maintenance operations in place as the benefits of a robust maintenance systems allow for more profitable and safer production operations. Over the years, maintenance ideologies have changed and evolved to address specific gaps within maintenance operations and each industrial application takes its own approach to imrpove production and reduce downtime to perform maintenance.

The following are high level description of maintenance in various industries and unique factors to them.

2.1.1 Maintenance in Manufactured Goods Based Industries

Maintenance in manufactured goods based industries aim to maintain assets involved in the production of manufactured goods with the objective of meeting production demands and performing the appropriate amount of maintenance on manufacturing system to extend their functional lifespan.

Depending on the manufactured good and its production demand, these industries can leverage downtime or buffer periods, in which the production asset's operation can be temporally halted. This allows for time period in which maintenance can be performed. Buffers are areas between different production stages where semi-processed goods are held awaiting for the next production stage. These buffers are created as different production processes take different lengths of time to complete and to create a safety in which production can continue if a prior process system is unavailable. These are particularly useful for maintenance planning as there are discrete maintenance periods that can be calculated based on production rate and buffer size. If enough buffer is created, maintenance on the system prior to the buffer can commence until the buffer is depleted and a production stop loss is established. This can be seen in Figure 1 as the three different production stages each have their own completion time and the buffer zones each hold a different number of components. Once enough buffer is established, Production Stage B2 can continue while there is an opportunity to perform maintenance on A1 and B1 until the buffer stock is depleted.

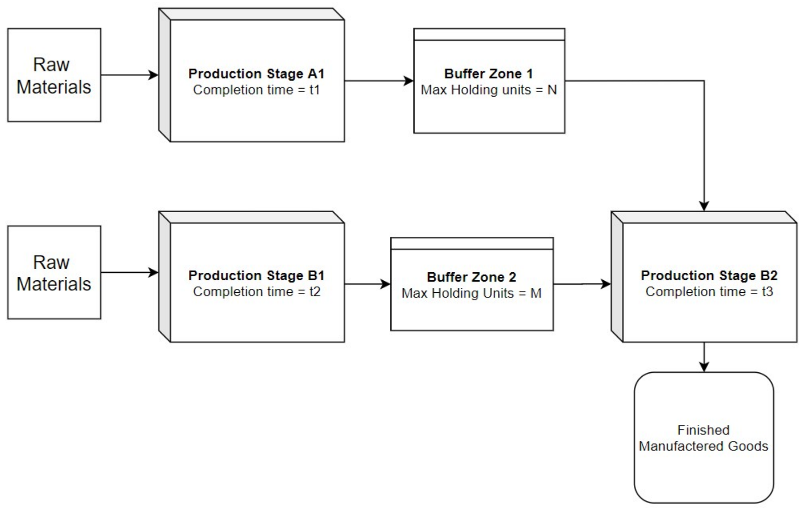

Figure 1: An example of a manufacturing flow. Each production stage has its own time of completion per unit followed by a buffer zone that each hold different number of in-process components.

These maintenance industries also vary compared to others in the fact that the production processes are generally isolated from one another. Meaning that the function of one processing system does not impede on the ability for a different processing system to operate.

2.1.2 Maintenance in Transportation Industries and Services

Semi-trucks, car fleets, airplanes and boats are crucial assets in transportation industries and services. These are similar to manufacturing industries as the usage of these assets are based on the demand but differ as the assets are either in service or not; their output cannot be lowered or increased. This changes maintenance planning as non-emergency related maintenance is to be timed for when an asset is not in use and maintenance must ensure that asset can last the entire duration of the trip.

2.1.3 Maintenance in Public Infrastructure

Roads, buildings and other structural assets require maintenance as well with their own specific requirements. Maintenance in structural infrastructure must be planned primary in accordance to their usage and safety.

Damaged infrastructure can often cause deterioration and damage to other assets such as potholes causing car damage. Maintenance in these scenarios are also more periodic than continuous. Infrastructure maintenance also must consider temporal factors. An example of this is road maintenance preferred during the night time. This is to ensure no traffic jams occur and it is safer for workers. These temporal factors may also alter the time between a damage conditions and maintenance execution.

2.1.4 Maintenance in Energy Production

Energy production systems such as oil and gas and especially nuclear have their own unique set of maintenance planning requirements. Generally, the differences arise in how maintenance tasks are delegated, safety considerations, and unique Original Equipment Manufacturer (OEM) considerations. These will be discussed in more detail in the following sections.

2.2 Maintenance Types

In all industries, different maintenance approaches are developed to address various gaps in the maintenance operations. There are 2 high-level categories of maintenance ideologies, Preventative Maintenance and Corrective maintenance.

Preventative Maintenance (PM) is an approach to maintenance where components and systems undergo routine maintenance activities before they fail in order to extend their lifetime and reduce costs. By performing continual maintenance over the lifetime of a component, it reduces the likelihood of failure and subsequent replacement of such components which generally costs more than repairs.

Corrective Maintenance (CM) can be defined as "the actions required to mitigate the consequence of a component failure and/or to repair or replace failed components" [7]. This maintenance approach essentially lets a component either fail or reach undesired performance standards until maintenance is initiated. Corrective maintenance (CM) does not include design changes and replacement activities as they are dependant on the specific component and the organization's specific approach.

Some of the most popular and in practice maintenance methodologies are presented in Table 5

Table 1: Types of Maintenance

Preventative Maintenance	Corrective Maintenance
Time-Based Maintenance	Planned Corrective Maintenance
Failure Finding Maintenance	Emergency Maintenance
Risk Based Maintenance	Deferred Maintenance
Condition Based Maintenance	Opportunistic Maintenance
Opportunistic Maintenance	

2.2.1 Time-Based Maintenance

Time-Based Maintenance (TBM), also known as Periodic Maintenance, is one of the oldest and most widely adopted PM methods used. It relies on the use of failure-analyses from both internal and manufacturing recommendations to develop fixed calendar maintenance intervals [1]. Maintenance tasks are performed at these intervals regardless whether maintenance is required or not. This approach to maintenance relies on understanding the Mean Time to Failure (MTTF) and Mean Time Between Repair (MTBR), which are metrics that help determine the reliability of component, and developing a comprehensive schedule around it.

The benefit to TBM is that for known performance degradation issues in a component, the approach is very good at forecasting schedules, resources and associated costs [1]. Also, using various statistical methods such as Weibull distribution models and bathtub curves, the maintenance decision making process is streamlined as it defines the probability of failure based on where the component is in its life cycle [1]. The failure rates are dependent on 3 general phases, Burn-in, Useful Life and Wear-out as seen in Figure 2. The goal of TBM is to reduce burn-in, extend useful life and replace during wear out.

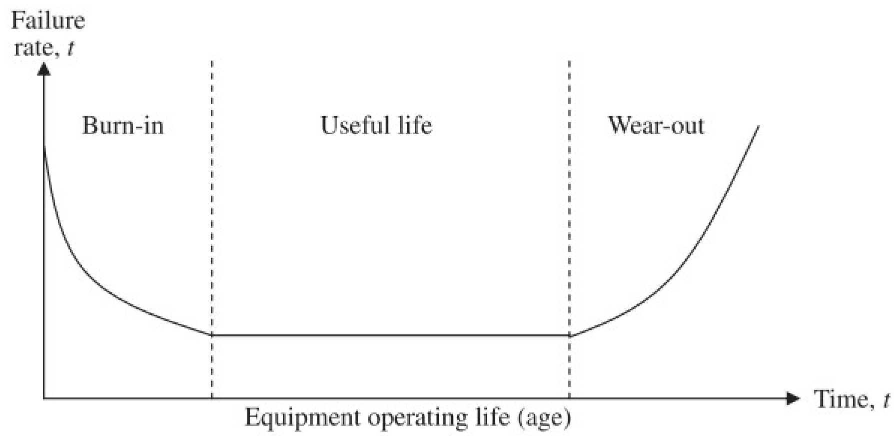

Figure 2: An Example of a Bath Tub Curve with the 3 phases (Burn-in, Useful Life, Wear Out) over the course of a component lifetime in relation to failure rate [1].

However, there are some issues with TBM that do not make it optimal for aging phenomena. TBM relies on multiple assumptions that do not hold up as a reactor ages and are better suited for gross generalizations in maintenance forecasting. One of these assumptions is that failure phenomena are predictable [1]. However, in nuclear reactors, many of the components undergo operating conditions and are commissioned for periods of time unique to the industry and are not as vastly studied compared to their non-nuclear industrial counterparts. Many industries such as nuclear and hazardous material storage have changed their approach to risk assessment due to the gap within the knowledge space [8]. It is now understood that components can be subjected to 4 categories of failure as follows [8]:

1. Known Knowns: Risks that are understood and can be managed with high certainty,
2. Unknown Knowns: Risks that an organization knows is possible but do not understand its the relevance to the asset,
3. Known Unknowns: Risks that an organization is aware of occurring but does not completely understand its mechanics, and
4. Unknown Unknowns: Risks for which neither existence nor mechanisms are understood.

Thus, predicting their lifetime using conventional methods leads to uncertainties as only Known Knowns are sufficiently understood. This also leads into the issue of imperfect maintenance. TBM assumes that maintenance returns the component back to its expected operating conditions however maintenance does not always bring back a component to its full potential use and this affects the validity of the statistical analyses performed [1].

Another issue with TBM is the lack of optimization. The way maintenance schedules are developed means that maintenance must occur at fixed intervals even if the component does not warrant it. In the nuclear industry, due to safety reasons, often entire systems need to go offline in order to perform maintenance on a singular component. This shutdown is directly related to lost profit and safety / redundancy. Thus, it is possible that the long term costs benefits of TBM are diametric to the savings in improved reliability.

2.2.2 Failure Finding Maintenance Maintenance

Failure finding maintenance (FFM) is a version of TBM that considers inspection data within its analysis of MTTF and MTBR. Within the maintenance schedule, there are inspection periods where a crew evaluates the condition of the component and compare it to criteria to determine whether maintenance is warranted in the upcoming cycle. Another proponent of FFM is failure discovery. Within these inspection periods, measurements are taken on various aspect of the components. This is done in hopes to find any alterations to the physical aspects of a component and performance. If there are behaviours that are not expected, then more exploratory and monitoring procedures can be added.

The benefit to FFM is that it addresses the issues in TBM where maintenance actions would be initiated even if the condition did not warrant it.
This maintenance approach is also beneficial as the fault discovery actions can help update the MTBR and MTTF metrics for more accurate long term maintenance management.

However, FFM only addresses some of the issues present in TBM. One of the biggest issues with inspections is that they are also periodically done. Inspections are usually set by industry experience with a level of conservatism however, the biggest area of concern are faults and failures that arise between inspection periods. Different industries have different approaches to inspection periods. In manufacturing industries, generally inspections can be done more frequent however, in nuclear many of the systems are embedded within each other and access for inspections is difficult. The frequency and degree of inspections are an ever evolving conversation within the nuclear industry due to this reason [9].

With respect to maintenance, periodic inspection is not the ideal scenario for maintenance planning as it only tells the state of a component after an event occurs and future events are postulated. For better maintenance management, determining and diagnosing faults before they occur helps determine level of effort, resources and ultimately cost.

2.2.3 Risk Based Maintenance

Another improvement on TBM was the introduction of Risk-Based Maintenance (RBM). RBM is a maintenance policy that prefers the placement of resources on the most risk prone components. The benefits of this approach include a primary focus on minimizing catastrophic failure events, both safety and cost related [10]. RBM should be seen as an improvement of FFM as the same principles of inspections are applied however the outcomes of these inspections change the focus of maintenance planning.

2.2.4 Condition-Based Maintenance

Condition Based Maintenance (CBM) is an approach that uses data from the continuous monitoring of engineered systems to determine optimal maintenance decisions [11]. CBM is comprised of three major stages, data acquisition, data processing and decision making [11].

1. Data Acquisition is the process of obtaining and storing data from various components. Obtaining the data is usually done in the form of sensors that monitor a systems operating behaviour [12]. The types of sensors used are dependent on the aspect of the system that is trying to be monitored. For example, infrared thermometers are used for temperature measurements and current sensors are used to monitor electrical flow and generate a corresponding signal [12]. The storage of data is done by transmitting the sensor information to a memory location on a computer or a server infrastructure. However, with recent advancements in technology, a more ideal approach for data acquisition is the use of Internet-of-Things (IoT) devices, wireless sensor networks, and cloud storage. Wireless sensor networks (WSN) use low powered IoT sensors

and transmitting devices set up in a network array to communicate to a base station to monitor an environment [13],[14]. As the sensors in these networks have evolved, the acquisition has as well. Cloud storage lets the information received from these sensors to be uploaded to offsite storage locations managed by external service providers such as Google Cloud and Amazon Web Services. The initial storage infrastructure is not needed on site and the information can be accessed anywhere with an internet connection.

2. Data Processing in condition-based maintenance involves the cleaning, managing and computation of the data from the acquisition phase. Data from the WSNs are not always reliable and can be subjected to quality issues [15]. As such, the data needs to go through various routines in which incorrect data is either removed or fixed to represent the accurate status of a system. In tandem, the data must also be organized in a manner that is usable by both the individual trying to read the data and the system trying to process it further. The computation of the data involves the statistical analysis to determine the current state of the system and the projected future state. At this point, data analytic techniques such as machine learning are used to predict failure times of components and optimal times to perform maintenance [16]. Other models also include the detection of failure modes to better diagnose system concerns. These models leverage the history of data collected to forecast when maintenance should be performed in respect to the current operating conditions.

3. The decision making in CBM is the analysis of the outputs from prior steps to determine the optimal time for maintenance. By knowing the current health status of an asset and the projected time to undesirable performance, allows for maintenance planning to commence at an earlier time with a higher degree of certainty. The goal is to gain the maximum usage of a component and utilize the resources as best as possible [16]. Here, the type of model is dependent on the type and amount of data available.

Pros: One of the major benefits of CBM is the better utilization of personnel. The earlier onset of planning allows for a reduction in lead time [17]. Another benefit is the earlier detection of faults and potential failures [17]. As CBM uses extensive amounts of data, any delta between expected operating parameters and true operating parameters are noticed earlier and preventative or corrective measure can be taken. CBM also provides a cost benefit over conventional time-based maintenance. In a time-based approach, regardless of how much useful life a component has remaining, the maintenance work will generally commence at the scheduled maintenance interval. CBM allows for operations to continue until an optimal cost related time for maintenance is reached [17]. Effectively reducing the number of maintenance periods (down time) over the life of a component.

Cons: However, there are a few issues with CBM which are currently being researched. One of the issues is related to the IoT sensors and the associated WSNs. Many of these devices are battery operated as they are embedded in areas that cannot be supplied with direct power [18]. As such the devices need to be able to last a minimum of a single maintenance period. This is related to other issues where each sensor requires a minimum reliability. The increase in the number of monitoring equipment increase the number of failure modes. Also, wireless sensor networks are known to have issues with coverage and connectivity. Environmental factors, the geometry of placement, the number of sensors and interference from adjacent systems can all play a role in decreasing the effectiveness of a WSN [19]. This poses a problem in transmitting the data to the storage location. Also, CBM is best utilized in situations where the failure modes are known. This allows for the correct sensors to be installed and the correct analysis to be performed. This leaves a gap in detecting unknown failure modes as the mechanisms would not be understood enough to place the appropriate sensors to gather the correct data. There is work currently being performed to account for this [20]. Data and the prediction algorithms can also serve to be a potential downside to CBM and will be discussed further in the following sections.

Field Data vs. Field Inspection There is a common misconception that CBM methods are used to replace field inspection data. This is not the ideal approach as CBM serves to provide more insight on existing knowledge. As mentioned, the inclusion of many sensors adds more points of failure in the monitoring system. As such, field data is important to verify that the WSNs are functioning as intended. Also, the instrument data serves to provide numerical analysis on the component but lacks in providing a visual analysis. The benefit of field inspection data is that it can enable certain algorithms to be used during the data analysis portion of CBM. Supervised learning is a process where the training on a machine learning model is done based on labeled data and the inclusion of field data would allow for a human guided approach where the field data is used as a verification of input data [21]. This may also be required by regulations depending on the industry. Many existing maintenance policies have inspection periods within them to verify the condition of the system [22]. For added confidence this implementation is still recommended.

Maturity of Approach Though CBM is a promising approach to improve system performance and maintenance planning, it is still relatively in its infancy. CBM in theory is excellent at addressing the gaps that traditional time based or risked based maintenance approaches have. However, there is a lot of uncertainty upon the implementation of CBM systems. Costs of implementation and technical viability on a large scale are not completely understood. Also, CBM currently is better in aiding the decision-making process for individual system components that are operation critical. A more developed version of CBM would consider multiple critical and auxiliary components in its maintenance policies to reduce costs attributed to down time. Another area of growth for CBM is in the algorithms used for the analysis and predictions. A lot of the proposed algorithms available are implemented with controlled lab settings and are trained using existing available data sets. Though the results are promising, there is not much work available presenting the performance of them in industrial applications. This uncertainty and the experimental nature of implementing IoT sensors and WSNs in industrial settings drives up the capital costs.

Practicality and Cost Though the benefits of a successful condition-based maintenance system are high, there are a few challenges in which implementing such a system is not practical. First, the component / system should have some sort of performance metric that can be monitored [23]. Meaning that CBM is not ideal for most static systems as there is not enough variation in their function in which condition-based maintenance provides a meaningful cost benefit over ordinary periodic maintenance. Also, CBM is recommended for critical components [23]. Critical components are ones that result in the biggest downtime if their function is inhibited, have significant costs or affect safety / reliability of the system. The costs associated with CBM are made up for with the extended lifetime of such critical systems whereas as non-critical components are generally more accessible to replace if failure were to occur. Also, the time from fault to failure needs to be understood well before implementation [23]. For various components if the time between detecting a fault and the subsequent failure is not sufficient enough for remedial actions, the CBM approach is not ideal because it does not allow for preventative measures only detailed diagnostics. One of the aspects that needs to be considered is the complexity of the system, system geometry and timeline of operations. Ideally, implementing a CBM system is the easiest and cost effective when the system is being set up for the first time or during an expected planned outage. This reduces the amount of lost time.

Summary of Condition Based Maintenance in Nuclear Power Plants As beneficial as CBM is, there are considerations specific to nuclear power plants that can limit its application.

Safety: As mentioned, a unique aspect to the nuclear industry is the tight margin of safety. Due to the nature of the technology, it is imperative that safety is a top priority. Ayo-Imoru and Cillier's work compared traditional nuclear simulators to CBM Artificial Neural Networks to predict transients and fault detection [24]. They identify that any alternative to the accepted fault detection methods must be able to compete economically and be safer [24]. Meaning, the detection should be quicker and cannot compromise on accuracy. Their work concluded that the CBM approach is possible to detect anomalies with better accuracy but the neural network errors need to be systematically reduced. In their case, the error reduction was possible however, the CBM approach cannot be considered in situations where the error cannot be minimized as it violates the safety fundamentals.

Reliability: Another concern that arises with wide-spread CBM usage is the reliability. Machine Learning approaches are very good at determining trends based on data that is witnessed during training. However, one of the biggest safety concerns is spontaneous fault conditions or *once in a lifetime* events. These are events that are part of the unknown unkwowns failure category and there is a lack of work in determining if CBM approaches are able signal these events.

2.2.5 Planned Corrective Maintenance

Planned Corrective Maintenance (CM) is when maintenance on an asset occurs once a known failure or severe performance degradation occurs.

The benefits of CM include the lower cost in planning and reduction in intermediate maintenance tasks. CM is beneficial in certain scenarios such as, tasks with quick ”return from failure” fixes, replacement components are readily available, non-critical components, and low demand components [24]. However, complex systems, non standard designs and high demand / availability caused the shift toward PM.

It was found that PM extends the life of a component and the costs of failure generally outweigh the cost of planning and execution of iterative maintenance tasks. However the costs benefit of PM over CM is heavily dependent on the effectiveness of prognostic systems in place [24].

CM can be categorized into 2 further instances of either planned and unplanned maintenance. Planned CM is when the organization acknowledges and understands the potential failure of a component within a specific period of time. The maintenance planning in this scenario involves the acquisition of components for said time period and also preparing the maintenance crew. In a PM based maintenance operations, planned CM has taken on the form of Deferred or Opportunistic maintenance which will be discussed in the following section. Unplanned CM involves emergent work where sudden failures or performance depredations of a specific component take precedence over other planned maintenance activities. This will be discussed in the following sections.

2.2.6 Emergency Maintenance

Unplanned maintenance is the situation where various phenomena, human and operational, cause a component to reach a state of unacceptable performance or failure when not anticipated. These states then initiate some sort of maintenance response. Unplanned maintenance is not ideal as this is emergent work and the costs, resource requirement and other factors may not be accounted for. Depending on various factors such as reliability and availability, these maintenance tasks may take priority over pre-existing planned maintenance activities.

2.2.7 Deferred Maintenance

As mentioned, with aging reactors, the number of maintenance tasks are increasing due to various aging phenomena of components, increase safety requirements, and technological obsolescence to name a few [25].

However, there are resource, supply chain and risk management limitations which affect the rate at which maintenance can be completed in comparison to the amount of pending maintenance task. Due to this, deferring maintenance activities are a well sought out option as it prevents down time and reduces costs related to material and labour [26]. Deferring maintenance activities involves putting certain tasks on a backlog to allow higher priority tasks to occur first or to do them when there is an opportunity present (safety or financial).

The way an organization determines whether a maintenance task should be postponed is dependent on a component's Reliability, Availability, Maintainability, and Safety (RAMS) [2]. As seen in Figure 3, the degradation mechanisms of a component affect the reliability of a component which initiates maintenance protocols. The execution of said maintenance then affects the maintainability. The overarching proponent is the overall availability of a component. Generally, in nuclear power plants, the organization wants to make sure that component and a redundancy is available for safe and continual (profitable) operations. Degradation, availability, cost and reliability models are considered based on historical operational data to see what changes in maintenance plans can be made [27].

As maintenance tasks increase, the organization needs to determine what tasks take priority with their limited resources. This is beneficial to maintenance operations as the organization is able to look at the various tasks and determine what tasks are redundant or non-value adding and potentially parse them from the maintenance pipeline.

However, the downsides of deferred maintenance are that it can bring CM based approaches back into an organization. By deferring maintenance, the component's Remaining Useful Life (RUL) might be affected and shorten as a result. This could lead to scenarios where CM procedures need to be performed instead of PM. Also, deferred maintenance tasks are often rescheduled into other maintenance periods effectively creating maintenance strain [28]. Furthermore, the long term costs associated with deferring maintenance are not fully understood. However, other industries, particularly construction and road maintenance, have done extensive research on the long term costs of delaying maintenance tasks [29]. It was found that depending on the length of deferral, there may be significant long term costs. If one is to assume the same properties for nuclear power plant components and infrastructure apply, maintenance should only be deferred for a certain period of time.

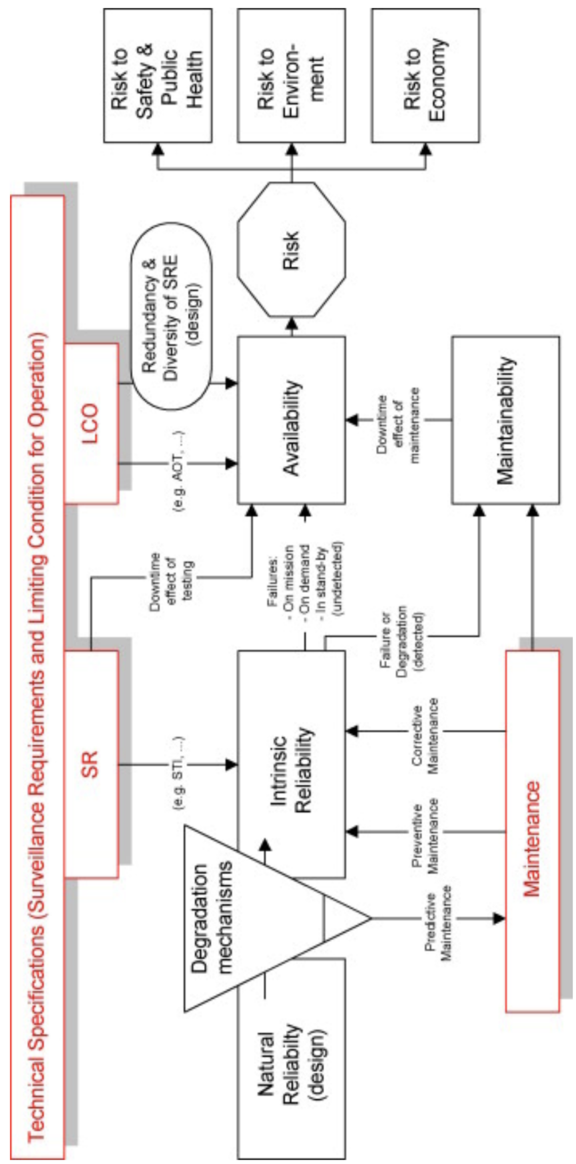

Figure 3: A flow diagram of a how RAMS relate to Maintenance Decisions and the potential risks associated as discussed by Martorelle et al. [2].

2.2.8 Opportunistic Maintenance

Opportunistic Maintenance (OM) is an approach that places a hierarchy of considerations for maintainable components and maintenance is initiated in a window of opportunity that best addresses the designated considerations. It is designed to optimize maintenance prioritization with respect to resources and costs.

Maintenance Priority In OM, one of the key aspects in developing an effective maintenance schedule is determining the prioritization of maintenance items. As there is a limited amount of time and resources available, it is essential to perform maintenance on components that are mission critical or provide the best return.

Within manufacturing industries, maintenance prioritization is a well researched field as it is directly related to downtime which is directly proportional to lost production and profit.

Tam and Prince presented a framework to highlight the critical decision making dimensions [30]. For asset management the 3 main dimensions are the following as according to [30]:

1. Output: Prioritizing dependant on whether the organization's production or delivery objectives been met. Examples include minimum operation requirements and maximum shutdown time.

2. Risk: Is the consideration of system reliability and risk of failure. The include safety, financial, and potentially social.

3. Resources: These considerations include human resources, facilities and tools and spare components. All of these considerations directly tie into budgetary restrictions.

Another approach to maintenance prioritization is based on the shutdown level required. Seif et al. presented work that showed a maintenance optimization method with the objective of minimizing equipment shutdown [31]. Their work specifies shutdown levels from 0 to 6. Level 0 refers to no isolation needed, Level 1 is local isolation all the way to total shutdown which is Level 6 [31]. Their work presented a way to minimize shutdown time by grouping maintenance tasks into opportunistic windows and removing redundant tasks based on shutdown level required.

There is also multi-grouping approaches where maintenance prioritization considers grouping tasks together based on safety requirements, functional deterioration such as common cause failures, economical reasoning and ease of maintainability. In nuclear power plants, an example of multi-group maintenance would be outage maintenance and refurbishment maintenance. In recent years, as outlined by Wang et al, grouping approaches provide significant decision making capability for manufacturing and power generating systems [32]. Advanced algorithms are being developed to include multi component and system grouping into preventative maintenance frameworks. Current group approaches are looking into considering economical grouping. Wang et al work presented a decision making framework where a preventative maintenance model from a unit level feeds into a *"Grouping strategy and related cost"* and *"Feedback"* stages where the individual components from the PM prediction are grouped in "aged based", "maintenance cost" and "shut down cost [32]. The grouping receives verification from a feedback loop to ensure that it does not contradict reliability predictions.

Maintenance Hierarchy A more formulaic approach to maintenance prioritization is Analytical Hierarchy Process (AHP). It is the process of mathematically determining which option from a selection takes priority over another. This approach aids in decision making and can be beneficial for determining which component in an system takes precedence for maintenance.

Bosco's work commented on how maintenance modernization and decision making requires an expanded life cycle consideration including pre-project phase and funding [33]. His work presented factors / considerations in creating a comprehensive decision model. The considerations include the following [33]:

- Affordability
- Budget
- Cost: Life Cycle Cost and Total Ownership Cost
- Safety

- Reliability
- Maintainability
- Availability
- Interoperability
- Supportability
- Maintenance Approach (ie, CMB, PM etc)
- Personnel

Seif et al. work used a maintenance schedule where each individual maintenance project and task has a hierarchy value associated [31]. If the two different projects have the same maintenance task coincide, the project with the lower hierarchy will not complete the task [31]. Higher hierarchy represents a project that is more pertinent or a project that has a higher scope. To complete the tasks in the higher hierarchy project, the task from the lower hierarchy needs to be done regardless so it reduces doubling on a single maintenance item.

In a mathematical approach hierarchy is important as a well defined hierarchy establishes the conditions for optimization. An example of this is that safety systems will rank availability higher than cost but an auxiliary support valve may rank maintainability over cost.

Maintenance Improvement However, with prioritization and hierarchy conditions developed, an approach to calculate an optimized maintenance schedules is needed. Seif et al. work used Mixed Linear Integer programming that used operational constraints such as shutdown level, and resources to optimize the a schedule for oil and gas plant [31]. These improvements are what bring upon the opportunistic windows on which it is best perform maintenance on.

The positives in this approach is that it provides a measurable objective on which to improve the maintenance operations. Another benefit is that it allows for maintenance improvement depending on the specific requirement for a particular asset rather than a one system approach. However the issue with this maintenance approach is the optimization with regards to multiple components. Different components may vary with their priority and can cause conflicting schedules.

2.3 Maintenance Periods In Nuclear Power Plants

A maintenance plan is a document all components / systems have that dictates, the maintenance work to be done, assets required, coarse timings of when tasks should be initiated and the type of skilled workers needed.

These documents are established before the installation of any component and are regularly updated to consider new operating conditions and constraints. These documents are developed using information from multiple sources including, manufacture recommendations, standards, guidance and policy documents, regulatory requirements and field data.

In nuclear power plants, these documents also include information on the organizational roles and responsibilities towards maintenance work, operating conditions to which maintenance is initiated, restorative states, and tasks separation by specific maintenance period.

For various components and systems, maintenance tasks are separated based on different maintenance periods as described in the following sections.

2.3.1 In Service Maintenance

In-service or On-line maintenance refers to maintenance work that is done while the reactor is operational and producing energy. Generally these tasks are more service related, periodic, and related to more accessible components of the power plant.

The minimization of in-service maintenance activities is preferred due to safety reasons as there is a higher risk with performing maintenance when the power plants systems are online.

Also, for many maintenance activities, certain components and systems have to go offline while the reactor is operating and this reduces the availability which is not a preferred condition. This serves to be a scheduling problem as well since maintenance of redundant systems cannot be done in parallel to potentially save time and cost.

2.3.2 Outage Maintenance

Outage maintenance is broken into 2 different categories. The first being scheduled outage and the second being forced outage. For the purposes of this work, only scheduled outages are considered as forced outages have different regulatory conditions.

The Nuclear Regulatory Commission (NRC) of America defines outages (scheduled) as the shut down of the generating unit and other facilities to perform maintenance, inspection and refuelling which is planned well in advance.

Different types of reactors have different outage periods however, during these periods, larger maintenance tasks are performed. These maintenance tasks involve the servicing of components directly related to the core of the reactor and high powered systems as they are powered down and the reactor is often de-fuelled. This reduces the risk to personnel and also opens different maintenance execution strategies.

This period of work is planned well in advance and have strict timelines as the production is halted and extended timelines directly translates to lost profits. In the ideal scenario, outage maintenance period should be reduced as much as possible while being able to complete as many maintenance activities as possible [34].

2.3.3 Refurbishment Maintenance

Part way through the life of a nuclear power plant, the reactors and other systems are taken offline to perform refurbishment. These refurbishments extend the lifetime of a nuclear power plant by "modernizing and enhancing major equipment and systems to support long-term operation" [35]. Refurbishment involves the replacement of major components and systems as a whole.

Refurbishment maintenance also have varying degrees of work. There is major maintenance in which a dedicated refurbishment team plans and supports in the execution of these activities. However, there are also auxiliary components and systems in the refurbishment units that need to be repaired. These activities do not differ much from in-service activities other than the fact they are being executed as the unit is being refurbished.

2.3.4 Emergent Maintenance

Emergent or Emergency maintenance occurs when a critical asset has failed or performance is degrading at unacceptable levels. This initiates CM actions and must be scheduled in with pre-existing planned activities. These tasks are not planned for and are not desired as it affects the overall planning and depending on the extent of the maintenance required, will deter from other high priority tasks.

2.4 Maintenance Planning Considerations

Maintenance planning is a complex task and involves the management of multiple components and work groups in order to develop a comprehensive execution plan. This involves getting the correct staff, material and equipment all while managing risks, both safety and financial. The following are some of the aspects that must be managed for maintenance execution.

2.4.1 Resource Management

In maintenance plans, there is information on the type of trade required for different certain maintenance tasks. When managing maintenance tasks, there must be an appropriate number of crew members to meet work requirements. This also involves having crew members with the correct training and qualifications [36]. Such qualifications include trades licences, working at heights, confinement training etc.

As hiring and training processes require time, knowing upcoming maintenance work is important in order to get the appropriate staffing. Another consideration in resource management is having enough work scheduled for a future period of time. This involves choosing the appropriate tasks for a work period that accommodates the fleet of crew members. Having no work for crew members is wasted potential and having too much work results in unreasonable timelines and safety concerns.

This type of resource management is common in multiple industrial applications however there are multiple requirements specific to the nuclear industry. Due to the nature of the work, radiation dose is a highly important consideration. Every country has designated requirements for the maximum allowable dose and managing the dose acquisition is important as it can potentially limit the type of work and location of work within the plant [37]. Also, maintenance procedures may require radiation personnel, so their availability, scope of work and scheduling needs to be considered as well.

2.4.2 Supply Chain Management

An effective supply chain management system is important for any nuclear power station to manage the flow of components and systems. In regards to nuclear maintenance programs, supply chain management plays an important role in the commencement, execution and completion of various maintenance projects. Androjna et al. describes the supply chain management and nuclear maintenance relationships using the following characteristics [38]:

1. The management of services to be delivered during plant outages. Nuclear power plants tend to outsource a lot of maintenance work to 3rd party contractors. As such, it is important to manage the acquired services and effectively schedule them.

2. Many services are turn-key basis (contractor takes full responsibility of engineering and manufacturing).

3. Many of the human resource requirements are handled by the main contractor for added value to the customer. However, there must be a direct flow of communication between the customer and the main supplier on the actions to align with both organization and regulatory requirements.

4. The flow of orders between customer (station) and main supplier. It is important

to understand if certain work is being conducted using sub-contractors and sub-suppliers and manage them accordingly.

5. The heavy integration of regulatory requirements in both service execution and component specifications requires a specific flow through multiple parties.

A study was completed, that highlighted some of the general barriers in supply chain management systems [39]. Extrapolated to nuclear power plants, in specifically maintenance projects, one of the major barriers is managing the flow of contractors in a dynamic environment. Work does not always proceed according to the schedule due various reasons such as scope creep and these delays can alter project progression and the flow of services and equipment reliant on the delayed task. The primary objectives of supply chain management systems in a nuclear power plant would include the following:

1. Reduction of overstock for spare parts and where possible go towards an on demand approach.

2. Having sufficient time to procure components and services before they are needed.

3. Having accurate information on original equipment manufacturer and contracting agency status for future inquiries / work.

Another consideration for supply chain management with respect to maintenance planning is obsolescence of Original Equipment Manufacturer (OEM) or Diminishing Manufacturing Sources and Material Shortages (DMSMS) [40]. As many components are in service for extended periods of times, sometimes up to decades, come time for maintenance service or replacement, if the OEM is no longer in business or technological advancement prevents the procurement of the required part, the component needs to be sourced elsewhere and go through design verification process. If this process is not initiated in time, the delivery of the component to the maintenance crew is delayed, effectively delaying maintenance.

Understanding the work scope is also important for the supply chain management to interface with the maintenance management. The work scope dictates the exact material amount and type needed and for the task. As a result scope creep or extended work can result in maintenance delays if there are not a stock of material readily available or procurable which in result affects the availability of component / system, delays work progression and subsequent tasks.

2.4.3 Risk Management

As with any business decision, maintenance management comes with risks that need to be considered when planning. The following are some of the risk considerations when planning for maintenance.

Radiological Risk Performing maintenance in nuclear power plants can subject crew members to higher than normal levels of radiation as prescribed by the country's guidelines. One of the considerations is the accumulation of radiation at various stages of the maintenance execution. Planning must consider factors such as, exposing radiation when opening up a component and dose from adjacent components and systems in both in-service and outage maintenance. The radiation risks would alter how a crew would approach the work including additional personnel to reduce accumulated dose, procuring special equipment to safely work, training and what maintenance period should the work commence under. These risks and their associated considerations are discussed in regulatory guidelines, procedures and requirements such as Department of Energy's *Nuclear Facility Maintenance Management Program Guide for Use with DOE O 433.1B* [41].

Safety Risk Similar to radiological risks, conventional safety risks such the ones outlined in Occupational Health and Safety (OHSA) requirements are considered in maintenance planning as well. Power plants are intricate systems with a lot of components at extreme operating conditions and the general layout presenting various spatial hazards. Due to this, there is a conservative approach in overall operations and management creating a high standard for safety culture in the industry. There is a heavy emphasis on creating safe methodologies to work and enforcing them. This affects maintenance planning as the conventional risks can also alter procurement of safety equipment, training, and maintenance period.

Financial Risk Another major risk factor for maintenance are the financial risks. All the previously mentioned factors have some sort of cost value associated with them and every decision has a financial risk associated if execution does not go accordingly. The following are some of the decisions that hold financial risks:

1. **Maintenance Approach**

   The decision to let a component run-to-failure or going with a preventative maintenance approach has different costs. If a component is non-critical, a CM approach may seem financially better however there is a risk of collateral damage. Whereas a CBM approach may propose to extend the life of a plant however there is a risk if the sensor apparatus and algorithms do not work as intended.

2. **Maintenance Period**

   Different maintenance periods may hold different costs due to safety and type of work. For example, if some component is approaching its service date but outage maintenance is scheduled in few months, there may be a preference to delay the service until the entirety of the system is offline. However, there is a risk that the cost savings from deferring to outage may be offset from other factors such as component degradation and potential schedule congestion during outage.

3. **Resource**

   The staffing at nuclear power plants are either direct staffing from the utility provider or sub-contractors hired for specific jobs [42]. Refurbishment maintenance is a common example where large subcontractor networks are used

[43]. Depending on agreements with the vendors, the cost between direct staff and subcontractors may differ as well. Scoping the work correctly directly reflects the budgetary expectations when it comes to staffing as scope creep and inefficient utilization of resources incur costs.

4. **Supply Chain**

   Within the supply chain, DMSMS problems often create financial risks. Not having a ready supply of material an equipment can serve as a bottleneck in the work as new sourcing is required. As mentioned in Section 4.2 due to the service life of many components, the OEM may not be available to procure a replacement. This would require new procurement process which would entail verification and validation of a component. Sometimes a replacement component is not possible to manufacture and that creates a financial risks of design changes [40]. If the OEM is no longer available, supplier selection problems occur as well as establishing a new supplier relationship.

2.5 Maintenance Scheduling Improvements In Nuclear Power Plants

The following section presents how scheduling with a nuclear power plant occurs. Due to organizational differences the following are general practices within the industry.

2.5.1 Scheduling Approach for Different Periods

For planned maintenance activities, planning is done at various levels for the different maintenance periods In Canada, the Canadian Nuclear Safety Commission (CNSC) guidance recommends that planning includes; ”daily maintenance planning, planning for next scheduled outage, and planning for future outages” [44]. In Japan, nuclear power plants undergo technical evaluations and the findings are utilized to create maintenance plan for the year ahead [45].

Due to lead time required by supply chain for procurement, the way budgets are handled and to create scheduling milestone, In-service maintenance activities start planning approximately a year in advance using multiple periods of execution throughout the year. Between the start of planning to the execution date of maintenance there are multiple milestone dates for the various work groups to interface and to ensure the crew is ready to execute the work. Some of these milestone dates include deliverables such as finalization of work scope, design package for supply chain, final date for task inject, and schedules.

For outage maintenance, the planning and decisions are done according to the technology of the type of reactor. Pressurized Water Reactors (PWR) and Boiling Water Reactors (BWR) require shutdown refuelling approximately once every 2 years and outage maintenance occurs during these periods but reactor technologies such as Canada Deuterium Uranium (CANDU) reactors have online refuelling so their outage periods are maintenance specific and are based on agreements with the regulator. In Canada the planned outage cycles is around three years [46]. Similar to In-service maintenance, planning commences well in advance with specific milestone dates. The only differences is that outages have much more strict timelines and work scoping can commence as early as previous outage periods. In Canada, the utility companies have agreements for planned outages every 12 years with the specific intent of performing thorough inspections [46]. Depending on the findings of these inspections, planning for certain maintenance activities can occur multiple outage periods in advance.

Refurbishment maintenance is planned similar to in-service maintenance however there is more coordination with the work groups responsible for major refurbishment activities.

2.5.2 In Service Maintenance Reduction

Due to the increase in maintenance activities, there is a constant effort to try to reduce unwanted maintenance activities and one of the approaches is to reduce the number of in-service maintenance activities. As mentioned, in-service maintenance activities are planned in small periods of time. Some of the reduction methods are useful in improving the overall plan however other methods are due to limitations and are not idealistic.

Task frequency updating One of these methods is updating maintenance task frequency. Maintenance task frequency refers to TBM frequency of performing some sort of service work such as an oil change. The initial frequency at which a task was scheduled for generally reflected the best practices as suggested by the manufacturer. However due to performed maintenance, new field data, new operating configurations and changed costs, maintenance planning determine if the task frequency can be improved to reduce redundant work. For many systems that run based on CBM methodologies, the frequency of maintenance is dependent on the monitored conditions so maintenance is scheduled based on when the system says it is time do so. The goal is to reduce the number of times a singular maintenance task is performed as there is a low return on investment if done more than the optimal amount with respect to component reliability.

Task Removal Another method is removing tasks altogether. An an example of task reduction can be found by British Energy, who in the 1990s, performed a set of optimizations by removing maintenance tasks they deemed non-value adding with respect to reliability of the component [47]. The tasks ranged from "simple preventative tasks" to various "invasive maintenance activities" and differed from site to site [47]. Results showed that the removal of certain tasks did not affect overall reliability and opened up opportunities to perform other maintenance tasks, though some tasks were reinstated due to component performance [47]

Deferring Maintenance As mentioned, actively deferring maintenance is an approach that is being considered in the nuclear industry. Deferring is a result of prioritizing certain tasks over others due to limitations in resources and other planning bottlenecks. There are multiple periods to which certain activities are deferred. These include:

- Outage: Some tasks are deferred into outage due to the type of work, priority and financial considerations.
- Refurbishment: A lot of maintenance work is deferred into refurbishment due to

ease of access to the component or system and other factors similar to outage.

- Future In-Service Periods: Due to emergent work or issues occurring during maintenance planning, a lot of work is deferred into future in-service periods.

2.6 Effects of Deferring Maintenance

2.6.1 Maintenance Backlog

Maintenance backlog is a list of outstanding maintenance tasks for a given asset. These are tasks that could not be completed during their scheduled time but are still required in order to ensure the reliability of an asset.

Tasks within maintenance backlogs can range from safety related items or to tasks to maintain minimum performance requirements. Within nuclear power plants, maintenance backlog tasks are also categorized further as elective or corrective backlog tasks [48]. Elective maintenance backlog tasks are preventative maintenance tasks that have been identified to be completed when an opportunity arises. These tasks are generally to improve performance and reliability [48]. Corrective backlog tasks refer to general unscheduled corrective maintenance tasks as described earlier. These tasks are more safety or performance related.

Backlog Maintenance Indicators: Assessment of maintenance system performance of the organization or a particular asset is done by evaluating various Specific Maintenance Indicators (SMIs). The following are some of the SMIs used in evaluating backlog maintenance according to [49]:

- **Number of outstanding backlogs:** Is a numeric value representing the sum of backlogged maintenance tasks to be performed for either an asset or organization as a whole.

- **PM work order backlog trend:** Depicts the rate at which backlogged tasks are increasing or decreasing over a specified time period.

- **Ratio of corrective work orders executed to work orders programmed:** This indicator can represent the percent increase in CM tasks than initially anticipated due to emergent work. On the contrary, if the ratio is lower than anticipated, the performance indicator represents that fewer number of CM tasks were performed than planned, however this does not describe reasoning. A low ratio generally indicates more exploration needed to describe the circumstances behind this.

- **Overdue of preventative maintenance activities:** Refers to the amount and length of delayed PM tasks. This metric is specific to PM tasks as CM, more specifically emergent CM can take priority over PM work. This factor can indicate how much PM work is deferred.

- **Number of jobs planned but not performed:** This number indicates how many new items are considered to be added into the outstanding backlog indicator. The difference between this and outstanding backlog is that this metric is concerned about tasks within a given time period that are added to the backlog list whereas outstanding backlog typically represents the total amount of backlog. However, not all tasks within this category get placed into maintenance backlog as some tasks may be cancelled altogether.

- **Number of jobs not started as planned:** This task is more representative of the true number of tasks placed into outstanding backlog. Generally tasks not started as planned occur some sort of material, scoping or resource problem which delays and consequently defers them.

Applicability to Deferred Maintenance: Deferred maintenance and maintenance backlog will be interchangeably used within this work due to the similarities with respect to the framing of the problem. Both deferred maintenance and backlog maintenance tasks add to a list of tasks that are to be completed. Backlogged tasks can be seen as deferred maintenance items. Deferring maintenance is the act of purposely placing a maintenance task into a backlog.

2.6.2 Reason for Deferred and Backlog Maintenance

As with any industrial process, nuclear power plants are also subjected to scheduling issues. In various studies it was shown that scope creep also affects the plants ability to get critical work done. In plants such as Fermi 2 in Michigan, 196 activities and 92 critical activities were behind schedule of the 993 examined [50].

Scoping Issues One of the more common issues in nuclear power plants is scope creep, in particular, unintended scope creep which site to a major cost overrun factor in nuclear power plant projects in general [51]. Scope creep can come in many forms including, new work, higher damage than anticipated, and organizational differences.

Other scoping situations can include work shortages. For invasive maintenance tasks, if the found condition does not warrant maintenance, the scheduled period for maintenance work is now free.

Undetermined Deferred Maintenance For backlog maintenance, sometimes there is not a clear path forward on when to reschedule and the task becomes part of the maintenance backlog. Maintenance backlog is a constant issue in power plants and organizations such as Ontario Power Generation identify maintenance backlogs every year and as a result have business objectives to reduce this number every year [52, 53].

Maintenance Planning Another issue with scoping comes in the form of the maintenance planning milestones. General practices have a milestone date to which the work scope for a period is finalized. However, resource issues, supply chain delays, various risk managements cause scheduling changes after the defined milestone date. Within this short period, the existing tasks need to be scheduled and may not be the most ideal especially if a task is dropped or execution is delayed.

Emergent Maintenance and Improper Maintenance Due to the aging phenomena, a lot more components and systems are subjected to new failures or degradation modes. This consequently increases the amount and frequency of emergent tasks and as a result resources need to be reallocated.

Improper maintenance is another issue that results in altered degradation patterns and will be discussed in the following section.

2.6.3 Design and Financial Effects of Deferred Maintenance

One of the biggest reasons to perform maintenance on assets is to prolong useful life. In the nuclear industry safety requirements make it such that components and systems are designing with a significant margin of failure. This means that components are designed such that they operate in a safety envelope magnitudes higher. Another design consideration in the nuclear industry is that the end of life for a component is based on some sort of failure event.

As seen in Figure 4, by assuming that after a certain margin to failure, a particular failure event occurs. When an asset does not undergo maintenance, the degradation will lower the Margin to Failure over time. In this scenario the degradation is assumed linear but may vary from asset to asset. The benefit of maintenance is that it will restore the margin to failure by a certain amount. Through regular maintenance, the margin is regularly restored and as normal degradation occurs the life is prolonged by avoiding the failure event margin. As seen in Maintenance Activity one and two in Figure 4, the margin is significantly restored and at activity two, and the non-maintenance equivalent experiences the failure event. At Maintenance activity 3, the asset is approaching the the threshold of the failure event but is able to avoid it by initiating maintenance. As a result, maintenance was able to prolong the life of the asset by 15 years.

The initial slopes of these graphs are based on the manufacturer information and is the design degradation. However, the operation of the components can lead to variations which will change the decay rate. Figure 4 shows the design decay rates of an asset in orange and the retrieval of the design margin with regular maintenance cycles in blue. The design degradation is the rate which the manufacture tends to guarantee under the specified operations and are often conservative in nature.

However, the burn in and operations of these assets can vary the rate at which an asset fails and the expected margin may not align with the true margin at a given maintenance cycle.

Financially, performing maintenance is beneficial as the component usage is significantly extended and a replacement is not needed earlier on in the plant life cycle. Due to these benefits, majority of maintenance in nuclear power plants is time based.

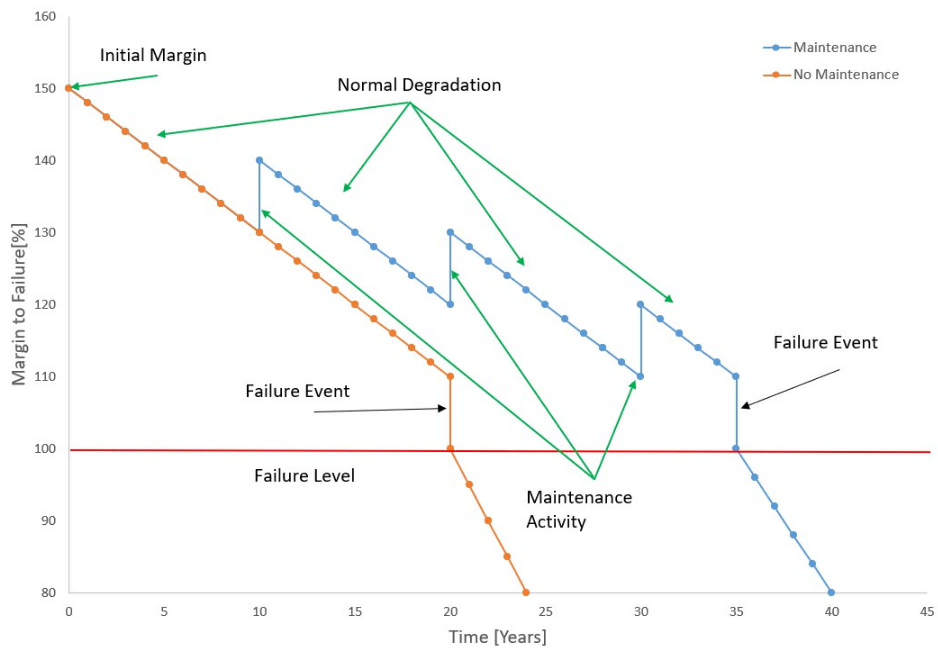

Figure 4: A graph representing the degradation of an asset over time based on performing maintenance and not performing maintenance.

However, the idealistic maintenance plan may not always be possible. As previously mentioned, delays and deferrals can occur which affect the life span of the asset as well as imperfect maintenance.

Imperfect Maintenance A common assumption in maintenance planning is that assets will return to a predetermined state once maintenance is complete. However, this is not always the case because minimal maintenance and imperfect maintenance actions are likely to occur due to the circumstances in plant [54]. In a period of high maintenance demand, the amount of maintenance for a particular asset is reduced in order to meet demand of other assets.

The more common form of maintenance is imperfect maintenance where "a maintenance action does not make a system like as good as new, but younger" [54]. However, many models suggested that the cost to bring an asset back to original margin is not optimal so allowable degradation models have been adopted such as the 10% improvement shown in Figures 4 and 5. However, imperfect maintenance is still a common occurrence as seen in the Deferred Maintenance Activities in Figure 5 where imperfect maintenance only brought up the margin by 8% rather than the anticipated 10%.

Imperfect maintenance is usually a result from many factors including but not limited to components not to spec, repair to designated component but cause unexpected damage to adjacent parts, hidden faults, and human errors [54].

Deferred Tasks In addition to imperfect maintenance, deferred maintenance can also cause issues affecting the life of a component. As deferred maintenance causes the maintenance task to be a part of the maintenance backlog, it is completed much later in the timeline. This does not necessarily cause problems immediately as seen in Figure 5 considering the two deferred maintenance items are performed when there is a considerable margin available.

However, with the imperfect maintenance being performed when more degradation has occurred due to deferral, the component is operating closer to the postulated failure event. This can be seen with the Emergent Maintenance task in Figure 5 where the asset is at a point in its life cycle where any further operations will fail it. In the normal maintenance model this point was approximately 2.5 years later. This can cause many downstream affects to the maintenance system as the time between two consecutive maintenance periods has shrunk from 10 to approximately 5 years. As it is Emergent Maintenance, it takes precedence and other tasks will now be backlogged which may potentially affect their Margin to Failure.

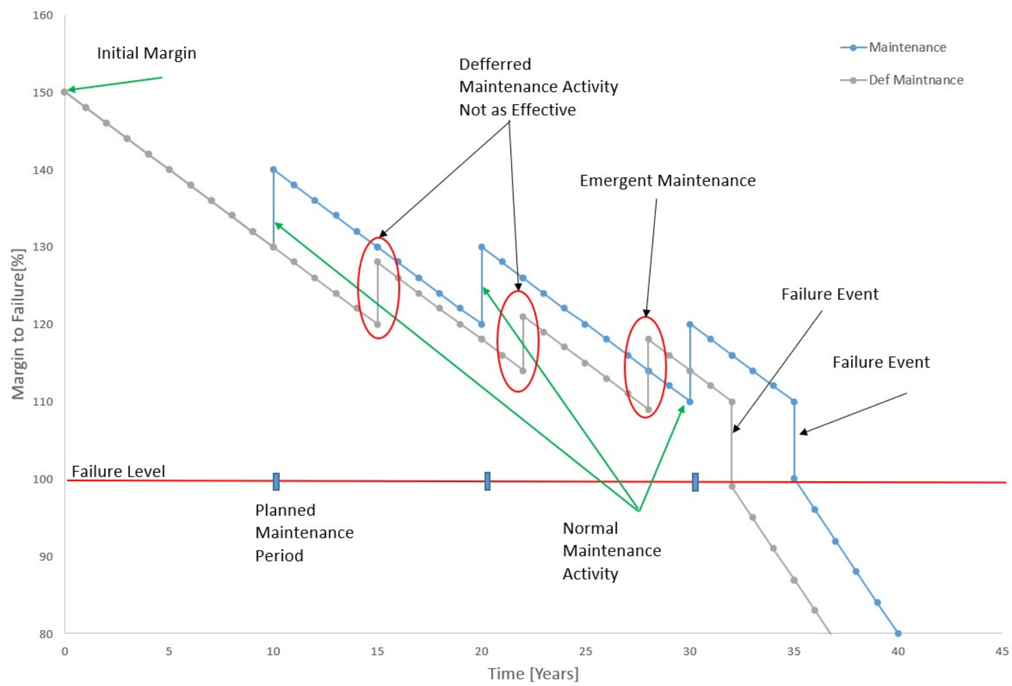

Figure 5: graph representing the degradation of an asset over time based on performing maintenance on time deferring maintenance.

From a safety and a reliability perspective, maintenance operations in this manner are not preferred. It is not reflective of best practices to operate components close to their safety limits when it is possible to control. From a financial perspective, deferral and emergent maintenance costs more than planned maintenance. Also, the earlier failure adds to the cost of the system as this is lost potential from the original asset that needs to be replaced .

As mentioned earlier, there are many risks associated with maintenance, however, with maintenance backlog, those risks increase and in many instances, additional financial and safety consequences can occur.

2.6.4 Additional Deferred Maintenance Issues

Rødseth presented some Key Performance Indicators (KPIs) for maintenance backlog as leading indicators and related them to profit loss KPIs which are lagging metrics [3]. These metrics are useful as they can aid in the understanding of backlog behaviour in an organization. A graphical representation can be found in Figure 6 found in [3].

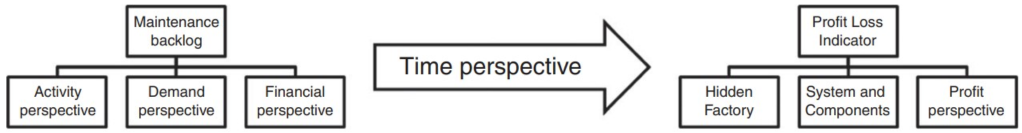

Figure 6: Leading Key Performance Indicators of maintenance backlog and their relationship with profit loss as a lagging key performance indicator as described by Rødseth [3].

The maintenance backlog KPIs are all measures of concepts discussed earlier and serve to be early indicators of maintenance backlog. The activity perspective is the measure of how much a component or system is online and operating whereas the demand perspective is the measure of how much a system or component is needed at a given time and the true supply. These indicators are useful for determining backlog maintenance because if either the activity decreases or the demand (output) is not being met, there is some maintenance that needs to occur. Financial perspective are a set of KPIs that assess the costs of running a component and this leading indicator can represent the costs of a component operating under non-ideal conditions due to it requiring maintenance and not matching the baseline KPI value.

However, these KPIs have relationships with lagging metrics that can help assess downstream effects related to profit loss. Hidden factory better known as Overall Equipment Effectiveness (OEE) is a measure of losses over a period of time with respect to indirect costs [3]. System and Component KPIs are costs related directly to the replacing or performing maintenance on a component or system. Profit perspective is an overall measure of hidden factory and system and components KPIs.

The one disparity in Rødseth's work is that Rødseth focuses on manufacturing industries where there is a direct measure of production and operations whereas in nuclear the effectiveness and cost benefit of systems and components are not as direct. However, the same underlying principle can be applied to nuclear maintenance when put in the context of component availability. Due to safety requirements a minimum availability for components must always be met and the rate at which a system is available can be analogous as production. Thus the following are the interpretations of the KPIs in the context for nuclear maintenance:

- Activity and Demand Perspective: This would be a direct measure of the availability of a system or component. The activity would be the status and health (Failure Margin) of a component at any given point and the demand would be the availability.
- Financial Perspective: The KPIs would assess how much it would cost to maintain an available component and the cost to bring an unavailable system back online.
- Hidden Factory: This would measure the total cost for maintenance on a particular component or system and the total amount of backlog that was unattended.

These representations of the KPIs presented by Rødseth's are beneficial for nuclear maintenance because often, the effects of backlog maintenance are not immediately known, thus making it harder to make informed decisions. By taking the lagging KPIs and relating them to leading KPIs, it would allow for active decision making and a better understanding in the effects of maintenance backlog. This formulation also represents why deferral is not ideal as there is a direct correlation between increasing costs and increasing backlog. As an asset deviates from planned safety margins, as described in the previous section, its performance will deviate as well. This will trigger a deviation in the expected activity perspective KPI which over time will signal an increase in the Profit Loss Indicator for System and Components.

2.7 Areas for Opportunity to Reduce Deferrals

Though deferral of maintenance tasks into backlog can serve for some benefit in the short term, there are long term affects which can degrade the overall maintenance performance of the organization. In order to improve on maintenance backlog and reduce deferral, some general improvements in maintenance operations can be made as followed:

1. Reduction of non-value added tasks [55],
2. Improving maintenance scheduling with respect to the organization's objective, and
3. Planning for maintenance risks.

Some additional specific improvements that can be are as followed:

Resource Forecasting One of the most important considerations when planning maintenance is human resources requirements. Most industries have a resource forecasting system in place to estimate the amount of skilled professionals needed for a given period. Many Computer Maintenance Management Systems (CMMS) have some sort of human resource management capabilities or are interfaced with dedicated resource management systems.

Under the umbrella of Machine Learning, this type of work would come under time-series forecasting and the applications have ranged from demand predictions for supply chain and general resource demand. In other industries such as cloud computing and IT services, they have implemented machine learning to predict the amount of computing resources required in a future time [56, 57].

In nuclear power plants, but not limited to, a potential room for opportunity includes creating a human resource demand forecasting model with respect to aging phenomena. As mentioned, resource limitations cause certain tasks being backlogged and under emergent conditions, human resources are deterred to other maintenance tasks. As maintenance work increases, using condition monitoring data from various components can help maintenance planning predict additional skilled workers required and provide a time period for human resources to find and train the appropriate resources.

Plant Layout Effects One aspect of nuclear power that makes it unique when compared to other industrial applications is the size and complexity of the plant layout. Unlike manufacturing facilities which are designed in isolated sections with each section having a different manufacturing process, nuclear power plants have extremely intertwined systems spanning various distances.

Some concerns that arise from this include the limitations on how long a component can stay offline, the extent of disassembly or shutdown required (local isolation to major system shutdown) and how maintenance procedures are sequenced (serial or parallel). A model to depict these considerations would aid in the reduction of backlog tasks because the level of effort can be better predicted going into a task, optimal work sequences can be developed and task grouping optimizations will allow for more opportunistic work windows.

Forecasting maintenance strain As mentioned, many maintenance tasks in nuclear power plants still rely on TBM approaches. There are thousands of maintenance tasks that need to be completed and all of them have different frequencies which may not synchronise with one another. This may lead to maintenance strain and promote backlogging certain maintenance tasks.

By forecasting maintenance strain, an organization is able reconsider its decisions and plan for any potential risks. In nuclear power plants some of the reasons for maintenance strain can include predicting if specific departments such as radiation protection are going to be busy in a given period.

Another importance in managing maintenance strain is to be able to balance unavailability and budget with respect to how much maintenance is performed. Wang et al's paper states there is a linear relationship between cost and the frequency and duration of maintenance, as defined as maintenance intensity [4]. However the relationship between maintenance intensity and unavailability of systems is non-linear. Both these can be seen in Figure 7 as found in [4]. Wang et al then go onto describe how high levels of unavailability is undesirable due to the safety risks and ultimately the financial risks associated. Thus, having a strictly financial based decision making process is not ideal as decreasing maintenance intensity can increase the probability of failure. On the other hand, increasing maintenance intensity can reduce failures but can increase unavailability because there are finite resources and more systems are offline due to maintenance increases, ultimately placing a strain on maintenance operations. There is an optimal amount of maintenance intensity which balances the lowest possible cost with respect to the lowest possible unavailability.

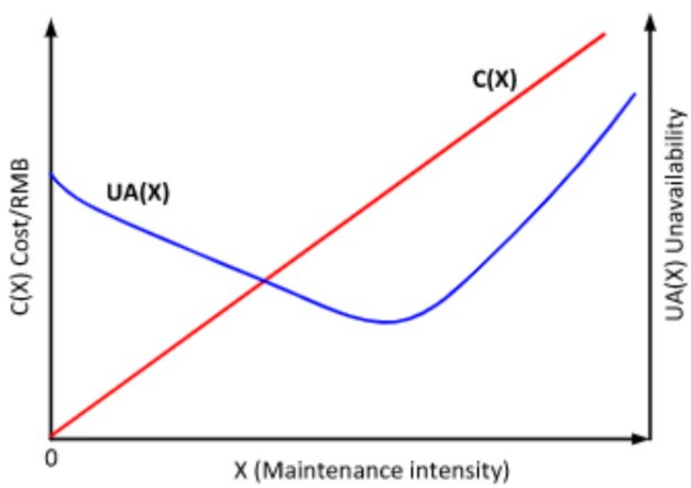

Figure 7: Chart representing the relationship between maintenance intensity, cost and unavailability as discussed by Wang et al [4].

Thus, having formulations on how to manage maintenance strain is important as it balances multiple considers such as cost and unavailability. This relates to the reduction in backlog because reducing the strain will prevent additional backlog and if there is room to increase the maintenance intensity, backlogged tasks can be scheduled in.

2.8 Maintenance Delays and Deferred Maintenance

As discussed in the previous section, managing maintenance strain is very important in the reduction of backlog. However, there are multiple factors that create maintenance strain. One of these factors is delayed tasks. As mentioned, there are thousands of maintenance tasks with varying frequencies and duration due to the majority of maintenance tasks following a TBM approach. Considering that many of the existing fleet of these power plants are halfway or in late-stage of their life cycle, many of these maintenance tasks have already gone through previous iterations and the maintenance management has previous experience on prior delays.

Figure 8 presents a representation on how compounding delays can cause maintenance deferral. As it can be seen, the top 4 tasks all incur some sort of delay, either in starting or execution. As a result, the resources were not available to complete task number 5 in the given time slot and the subsequent task is a critical task that cannot be delay. As a result, to ensure the resources are available for the critical task, task 5 was deferred into a backlog.

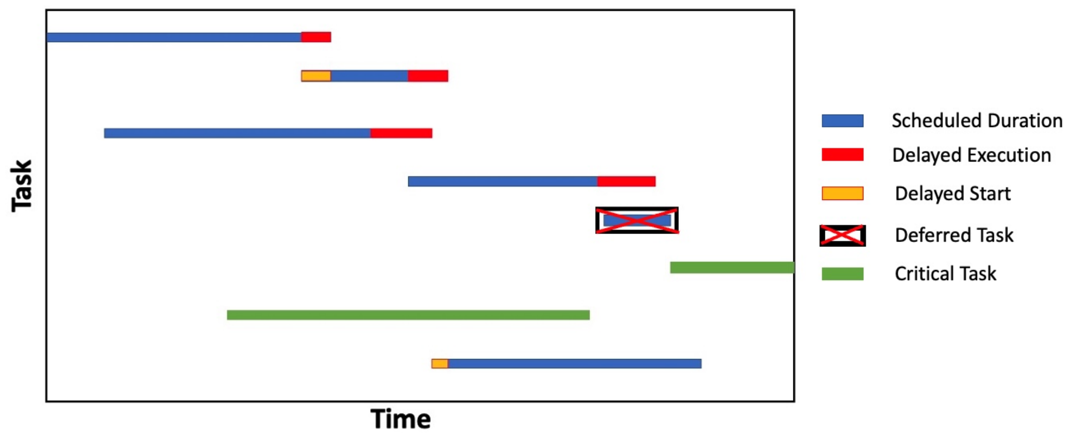

Figure 8: Graph representing how maintenance delays reduce resource availability, causing maintenance deferral.

The following are explanations on how some of the aforementioned maintenance planning factors create delays which ultimately leads to maintenance strain and deferrals.

Human Resources: If there are not enough workers available at the time of execution for a given task, work progression may slow down thus causing delays. This can include not only the maintenance team performing the task but others including radiation protection, and departments handling authorizations.

Supply Chain: If any materials, equipment or tooling are not available at the time of execution this will then delay the start of the maintenance task and subsequently have an affect on the completion.

Safety: Anytime during execution of a maintenance task, if the work conditions due to either radiation levels or conventional safety issues are over the safety threshold, the work will be stopped and investigated. If during the investigation it is found that the safety issues are not acceptable, the work will not progress until a method is found to mitigate the risk. The act of investigation and determine safe work approaches for emergent safety issues causes work stoppages which effectively delays the task completion. In addition to the work stoppage, if an altered work approach is required, this can slow down execution as it may be more conservative and a slower process, ultimately causing delays in completion.

Scope Creep: If a project is subjected to scope creep, the tasks may be halted until the asset's situation is assessed. This would require more effort and time to complete the task and as a result delays the completion.

2.8.1 Specific Factors Affecting Delay

Alsharif and Karatas work presented a number of causal factors resulting in delays for nuclear power plant projects [50]. The following are some of the more pertinent delay factors that they identified:

1. **Start Date:** When considering maintenance scheduling, start day is important as it serves as the date all activities, such as component procurement, needs to be completed by in order to execute maintenance. Also, the start day can help assess the overall maintenance effort for a particular period.

2. **Frequency:** This is the metric that presents how often an individual task needs to be completed. This is in reference to the asynchronous behaviours of maintenance as mentioned earlier.

3. **Duration:** Duration of the task is important in predicting delays as it dictates for how long a group of resources is busy.

4. **Human Resources Required:** This task represents the constraints on human resources in a given period. Is a limitations of simultaneous projects.

5. **Priority:** Certain maintenance items take precedence over other tasks within a certain period and this can result in other tasks getting delayed.

For nuclear power plants, there are specific considerations that alter the mechanics behind predicting maintenance delays. Some examples include the following:

1. **Shutdown Level:** Different components and systems required different level of isolation for safety or just accessibility. By considering the level of shut down required, this may be responsible for delays and would provide valuable context for the model to determine a correlation.

2. **Emergent Tasks:** Nuclear projects are subjected many scope creeps and if a project is a late inject or emergent work, this information will be important as the model can potentially use this as context for potential delays due to scope creep.

3. **Radiological Work:** The progression of radiological work is generally slower than other conventional maintenance work. As radiological conditions can halt work due to safety reasons, this information in the model can provide context on which works are more prone to delays due to radiological risks.

4. **Maintenance Period:** As mentioned earlier, maintenance in nuclear power plants is done in different periods such as in-service and shutdown maintenance. This input value is important because certain maintenance tasks can fall into multiple maintenance periods but only are subjected to delays in a specific period.

2.8.2 Maintenance Delay and Deferral Summary

Figure 9 represents a summary of the aforementioned factors affecting various stages of maintenance and their contributions to deferred maintenance. The section highlighted in red is the specific contributions of delayed tasks to task deferral.

Though this figure primarily focuses on nuclear power plants as a case study, the same flow logic applies to other industries. The only difference is the internal organization's importance placed on the safety considerations, emergent maintenance and resource availability.

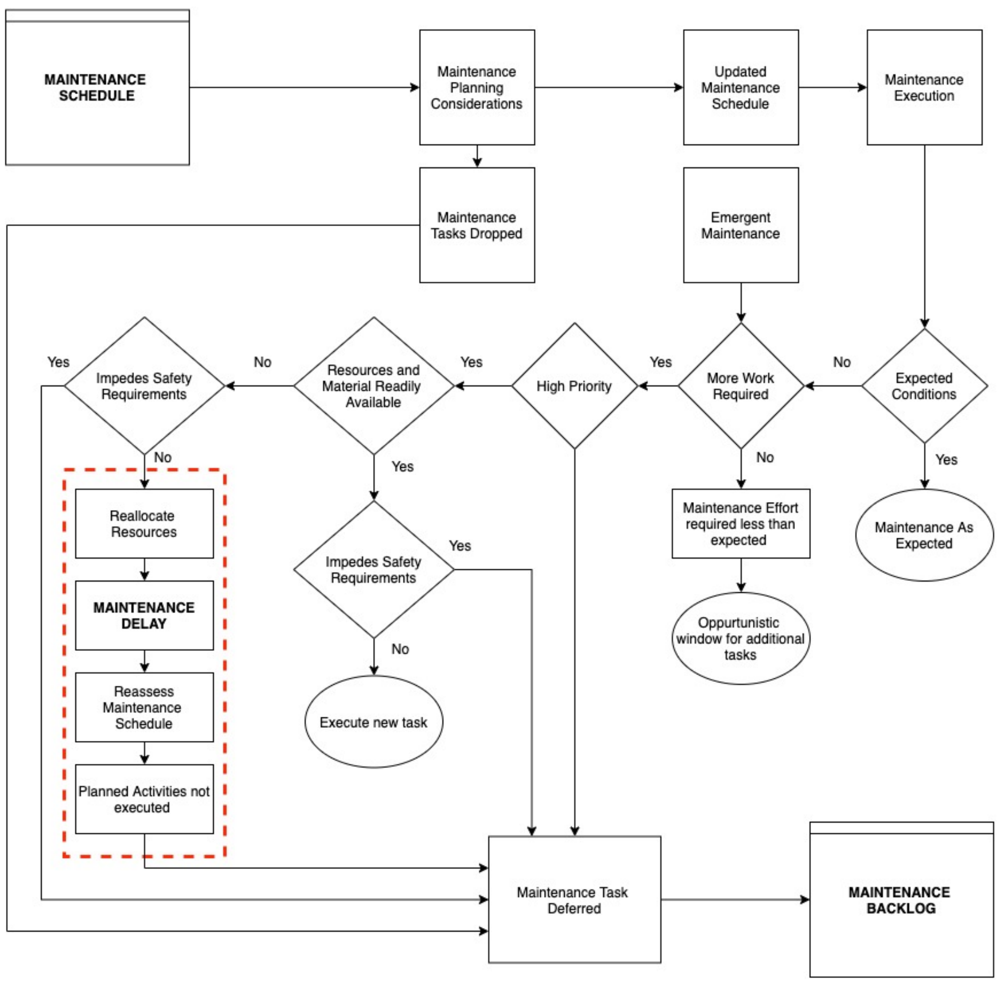

Figure 9: Flow diagram of how at various maintenance stages, maintenance deferral can occur. The section in red represents the focus of this thesis.

2.9 Delay Task Prediction

With the aforementioned information, a model can be developed in order to predict whether a planned task will be delayed and by how much. The value in such a model is that planning would be able to determine what tasks will place strain on maintenance operations and also would altering the parameters potentially mitigate some of the delays.

Also, by determining how much various tasks within a given time period are expected to be delayed, an organization can determine the cascading effects. For example, if a particular crew is scheduled for two successive tasks, by being able to predict that earlier task has a probability of being delayed, there can be strategies put in place to prevent the successive task from being backlogged. This is in line with Wang et al work as predicting maintenance delay is an indication of maintenance intensity [4].

2.9.1 Current Work in Delay Predictions

Currently, there are many works that try to predict delays operations in various industries. However, predicting maintenance delays is a relatively new area of exploration and a lot more delay prediction research is placed in different fields.

Many current works try to predict delays based on CBM approaches. With the use of IoT sensors and programming some sort of randomness to the system that is representative of conditions such as imperfect maintenance execution. These methods try to predict delays based on maintenance effort required. Some examples include Gbadamosi et al work on UK Railway maintenance, where they use IoT sensors for better condition monitoring of other assets [58]. Ayvaz's work presented using IoT sensors, to assess delays in manufacturing industries to better schedule maintenance [59].

Delay predictions are also used in various other applications other than maintenance. Truong's work presented a machine learning approach to predict the risk by airline delays which used a Baysien network to predict the probability a flight will be delayed [60]. In software development and maintenance, machine learning is being used to predict bugs so that delays do not occur in fixing these bugs [61]. This is very similar to nuclear plants using CBM however, reduction in delays is the focus.

Even though there are many areas of researching dealing with predicting delays, there is no current formulation of how to predict delays from historical maintenance records.

2.10 Current Software Systems

In order to determine a method on how to improve maintenance practices in nuclear power plants, it is important to understand the current data available and the current maintenance management system.

2.10.1 Data in Nuclear Powerplants

Data Sources: In nuclear power plants there is an abundance on the data available. Some of the primary sources of the data come in the following:

- Sensor Data: Operational data directly from the asset or system.
- Maintenance Logs: Data regarding the planning and execution of past maintenance items.
- Design Data: Experimental and reference data for how various assets should operate.
- Operating Experience: Data providing information on the lessons learned based on historical asset experience. Can be with regards to design, planning and operation.

In a typical power plant it is possible to find up to 10000 sensors for instrumentation and control with sampling rates of 1000Hz or 1000 samples per second [62]. Over the many decades of operations, there is a massive accumulation of just sensor data alone. Then considering that every asset has design, maintenance and operating experience data, there is a vast amount of data available to be potentially used for predictive analytics.

Data Availability: This accumulation of data is available to the utility that is generating it however, it is limited to their use. Due to immense cyber-security protocols and practices, it is preferred and often required for the data to stay within the utility's intranet rather than the internet [63]. As a result, very minimal published work is available on the data within a nuclear power plant. In addition to cyber security, competition is another factor for why data is not publicly available. This data is sensitive to the utility's performance and by releasing this into the public domain, would put the utility at a disadvantage. As such, most work on data anlytics for nuclear data is kept private within the organization and not publicly released. As a result, the exact contents and size of these datasets can only be estimated.

Data Considerations: In addition to the availability of the data, there are a set of considerations that may limit the ability to develop analytical methods.

1. Changing Standards: As mentioned, nuclear power plants can be operational for around 60 years. Due to this, plants have evolved through various data and information logging standards. As such data from even a few years ago may be in a form that is not readily usable for algorithms. This adds a complexity to implementing models as additional resources will be required to transform the data into consisting usable formats.

2. Data Management: As mentioned, there is a vast amount of data available and this can serve as a limitation as well. As the data is all internally kept, sourcing it and determining what information is available may add challenges because the industry has not gone through the rigour to manage the data in a method which is useful for advanced analytics.

3. Instrumentation Variance: Sensor data from components are known to drift over time and this introduces variances in the data which decrease the overall quality of the data. Certain sensors such as temperature sensors may have high levels of uncertainty to them thus, if it were a critical piece of information, the prediction quality of the model will not be as robust since the training data it is learning from has an inconsistent structure.

4. Severe Event Data: CBM approaches are also used to predict failure events. However, many assets in a power plant are commissioned for decades are critical components. Over the age of these components due to the stringent maintenance practices, there is no established history on potential failure events. Thus, there

is no data for the model to learn such events from. Though there is an abundance in data, the over all quality is low because it does not capture potential events for which CBM is designed for.

2.10.2 Computerized Maintenance Management System

Computerized Maintenance Management Systems (CMMS) are dedicated software system that contains all the data of an organization's maintenance operations and provides tools to track, plan and execute maintenance tasks. CMMSs generally are linked to the organizations asset management system in order to manage resources and also track tooling and equipment requests from supply chain.

Benefits of CMMS CMMSs allow for a consolidation of all maintenance data within an organization. It allows for the tracking of maintenance progress, spare part inventory a resource availability. All this information is digital and on demand while providing advanced analytics.

Cost Control A CMMS either allows for inventory control directly within its packages or allows to connect through external asset suites. This aids in managing inventory control and tracking supply chain requests. Automation in inventory reporting and purchasing can reduce costs. In addition, CMMS can track and report other maintenance costs such as down time and resource usage. Other cost benefits include the inclusion of condition based monitoring. This would aid in reducing corrective maintenance and improve the availability of components as mentioned earlier.

Increasing Efficiency Improving schedules is a major component of effective an CMMS. As these software provide in depth analytic tools, it is easier and more time efficient to make scheduling decisions as the calculations are offloaded to the software. In many cases, the system provides suggestions on best course of action if an anomaly is detected. There is also capability of automatically initiating actions given the right protocol. For example, if the stock of a general component is low, the system can automatically start a purchase order. The same initiating action can be used for work orders which improves access to resources. Standardization and consistency is another aspect of CMMSs that helps increase efficiency. As most of these systems are developed using industry experience, a lot of processes such as work reporting, work orders and condition reporting are embedded in the system and ensures that they are processed the exact same way every time.

Data Collection and Processing CMMSs are useful as they automatically calculate metrics pertaining to maintenance of individual components and maintenance operations as a whole. More advanced CMMSs directly feed in sensor data of various assets and eliminate the time users spend on data ingest.

As data is centralized in one location, this allows for easier access of historical data and many CMMSs provide trend analysis to determine maintenance performance. These trends help inform maintenance planning by tracking if maintenance objectives are being met and providing potential diagnostics if a problem were to occur.

Many modern CMMSs are incorporating condition based maintenance algorithms into their services so that the system can diagnose and automatically schedule new work orders.

Maintenance Backlog within CMMS Most current CMMS solutions address maintenance backlogs by providing tools to prevent it. These systems typically provide insight on why maintenance tasks have not commenced and provide suggestions for potential changes to avoid further backlog.

Rescheduling based on user constraints is already implemented in many CMMS solutions. Different systems use different methods for re-scheduling such as constraint based scheduling, mix-integer linear programming and machine learning based solutions.

There is also a lack of tools that allows for the user to predict and determine if maintenance backlog will occur beyond simple numerical checks if the required assets and human resources are available.

2.11 Application of Machine Learning to Delay Prediction

Due to the cyclical nature of maintenance in nuclear power plants and various other industries, the use of a machine learning may benefit in the prediction of future maintenance delays. Many power plants have logs of maintenance history including the delay, reason for delay, and other factors such as the ones mentioned in this thesis. In other instances, where historical data is not available, functional data of the asset can be used to derive analytics to understand the postulation of a maintenance delay.

As seen in Figure 10, a maintenance delay prediction model would help predetermine detriments in existing maintenance schedules. The model inputs would directly map to the maintenance planning considerations that can be altered to make the schedule better. A regression model would be more beneficial in this instance over a binary "delay or no delay" classification model because the extent of a delay can help determine the severity of a potential delay.

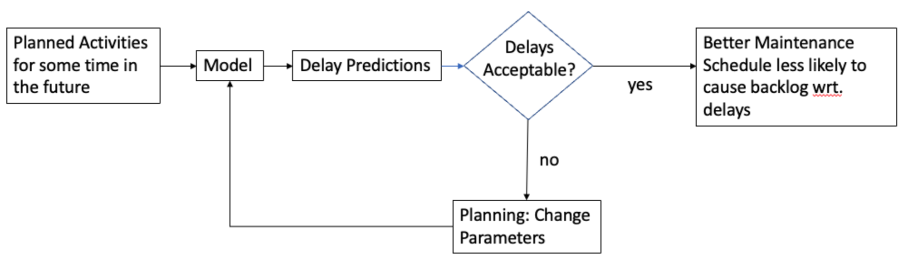

Figure 10: How a delay prediction model will fit into maintenance planning.

The work in this thesis is to develop a proof of concept framework(s) which could introduce a delay prediction such as this.

3 Data Driven Analytics

3.1 Machine Learning

Machine Learning (ML) is a data driven approach to understand and distinguish patterns in data that traditional approaches fail to interpret.

ML often uses Neural Network based approaches however, traditional statistical based approaches can still be classified under ML. The goal of using either traditional statistical or modern approaches stem from a goal to take data and find hidden patterns that can then be applied to new data.

An early approach to ML was formulated back in the 1950s with the creation of the perceptron [64]. A perceptron would take an input of variables, each multiplied by a numerical weight that represents significance of each of these variables, to then output a single binary value. However, the field of advanced ML did not catch momentum until the early 2000s where 3 different paradigm shifts in computation occurred, including Big Data, Parallel Computation and Deep Learning [64].

Big Data allowed for a mass accumulation of data in various applications in which more robust analytical approaches were needed. Parallel computing allowed for mass data to be processed simultaneously and efficiently. Finally, the ideas of the perceptron were advanced with new algorithms that leveraged the first two shifts in computation to what is now known as Deep Learning [65].

There are generally two classes of machine learning tasks, regression and classification. Regression is the process of predicting a numerical value from the input data whereas classification involves taking the input data and the goal is to bin them into predetermined class labels [66]. In this study, regression predictions are referred to as numerical predictions to prevent confusion between regression techniques which will be discussed later.

ML has a lot of different forms and applications in varying fields. Some of the more common applications of machine learning include self-driving cars, fraud detection, stock predictions and speech assistants [67, 68, 69, 70].

3.1.1 Types of Machine Learning

Another way to look at the different approaches to machine learning is to look at them as *Learning Tasks*. Due to the various applications and the different algorithms used, ML approaches are generally categorized into the following categories:

Supervised Learning: Is one of the most commonly used learning tasks in the field of ML. These tasks involve having labeled data to which the predicted output can compare to. In these tasks, a interpolation is generated to represent the input to output relationship with a number. Supervised learning is the focus of this work and will be discussed further in later sections.

Unsupervised Learning: These learning problems involve having data with no output variable and the model tries to discover patterns from the data. This is generally done by either clustering methods where data grouping is discovered or discovering data distributions. Unsupervised learning is powerful as it allows for dimensionality reduction to a space that is easily interpretable to the user. An example of unsupervised learning can be seen in Figure 11 where an unlabelled group of shapes is grouped by the learning task after determining which shapes have a curve or only flat edges.

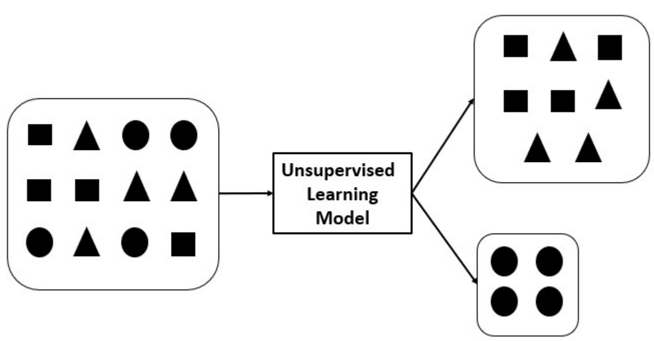

Figure 11: An example of a unsupervised learning task where the model is able to distinguish between shapes that have a curves or flat edges.

Reinforcement Learning: Is a learning task where the model learns a task on its own not through a robust set of data but rather well defined goals and objectives. In these tasks, an agent has a set of actions to chosen from within an environment. In each training loop, the agent picks an action to take and the state of the environment is recorded. Based on the state, a reward or penalty is assigned corresponding to the ability to meet the objective. From there, a new action is chosen and the training cycle continues. The training cycle is repeated until the agent is able to meet the objective with minimal error. Reinforcement learning is very popular in video games and self driving where the objectives and actions are well defined. An example of this can be seen in Figure 12 where the model is trying to learn how to score on the correct goal. There are two sets of actions each with their own penalty and reward. The cycle will loop until the agent is always able to score on the correct goal.

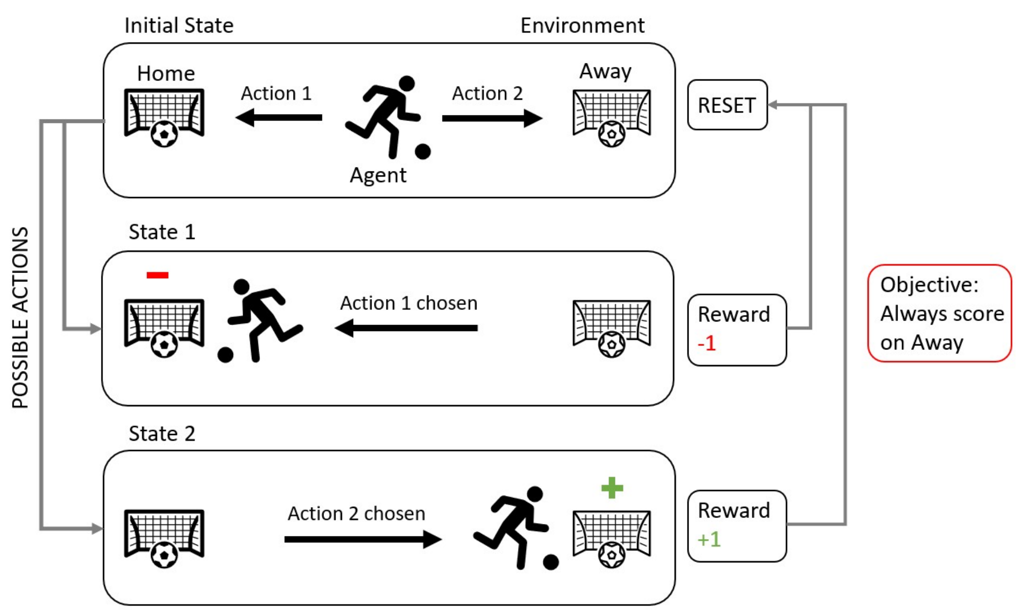

Figure 12: A basic example of a reinforcement learning task where model is being trained to score on the correct goal.

3.1.2 Parametric vs Non-Parametric Models

Parametric models assume that there are a set of numerical parameters that can be mathematically arranged to map the input to the output. The goal of a parametric learning task is to use training data to determine the value of these parameters.

Non-parametric models do not assume explicit forms on the functions relating the inputs to the outputs but try to construct mapping functions that are entirely dependent on the patterns in the data.

3.1.3 Data

The success of analytical approaches such as Condition-Based Maintenance is heavily reliant on the data collected. In order to have a model that is robust and able to predict with high accuracy, the data from which the algorithms learn from need to be high in quality.

Volume of data: The outcome of a CBM approach using supervised machine learning is dependant on the amount of data that is available for analysis. Traditionally in big data there are 3 major considerations regarding data which can be extrapolated for CBM. These are known as the Three Vs which are Volume, Velocity and Variety. The volume of data available is particularly important as it dictates how intricate the prediction models can be trained to which directly affects their accuracy. In CBM, more data also provides a better holistic view on the system and its performance to which maintenance actions can be taken if needed. Velocity refers to the rate of new data being injected into the system. This affects how much data needs to be managed as well as how often retraining or the rate at which continual training occurs. The faster data is being transmitted to the monitoring system, the faster the system condition can be updated resulting in more active maintenance planning. Variety of data represents the different types of data and the various sources. Generally a higher variety means a model can be trained to be more contextual and robust. Variety of data in CBM can seen as the measurements from the different sensors. The more variety in sensor types results in an assortment of readings that can be used to develop relationships in operating parameters. A lot of the fault discovery algorithms used in CBM find correlations within the data that represent the various operating parameters that are not obvious in traditional monitoring systems. A variety in data allows for the condition of the system and maintenance actions to be assessed based on different perspectives. For example, if detecting a potential crack in the pipe, by having temperature, acoustic and pressure sensors allows the system to determine if the cracks are occurring due to thermal stresses, vibrational stresses or pressure changes to best decide mitigation approaches. However, there is a cost trade-off by increasing the volume, velocity and variety of data. Increasing the volume of data available means there are more costs in physically storing and logging the information, the velocity of data increases computational costs and the costs associated with the commodities required to handle a high-bandwidth data transfer. The costs resulting in an increase in variety are attributed to the increase in sensors required to gather this data.

Quality of Data: As previously noted, during the data acquisition phase, erroneous data is a possibility due to potential issues in the sensors, the sensor networks, transmission to the storage location and servers. Common data errors include inconsistent readings, missing values, duplicate values, conflicting readings etc [71]. The importance in having high quality data directly affects the algorithms for data analysis and their respective accuracy [72]. Low quality can also result in excess noise in the data which would prevent the prediction from converging. The variance (error resulting from small changes in data) can change the sensitivity of the model [72]. Depending on the algorithm used, this could cause the CBM pipeline to consider system conditions that are not ideal to be considered as normal. From a cost analysis, low quality data increases costs of CBM. Low quality data is resultant in potential errors of the hardware, thus there are costs associated in remediating the sensor networks. Other costs would include processing and repair costs to fix data quality issues, increased IT maintenance costs, costs associated with retraining models and many more [73].

3.1.4 Pipeline of Machine Learning:

Machine learning leverages statistical methods thus there is no explicit mathematical formulation that ensures the model matches the data. Also, to ensure that model is performing correctly there are actions that need be taken.

Most machine learning implementations follow a pipeline from start to finish given a problem or objective is predefined. The following are the steps taken and a brief description. These will be discussed more in detail with respect to this work in the methodology section.

1. Data Selection: Finding or creating data that aids in the problem definition.

2. Preliminary Data Analysis: Perform data analysis to determine trends, correlations, outliers etc to have a better understanding on the data. The approach varies on the data being analyzed.

3. Data Cleaning: Upon the results of the preliminary data analysis, often times, there may be
components of the data that are erroneous and need to be addressed using various approaches. This depends on the data set size, type of error and extent of the error. Some common errors in data include missing entries, duplicate entries, inconsistent formatting and incorrect entries.

4. Feature Engineering: Depending on the type of attributes in the data set, the data may need to be altered. For example, categorical attributes need to be encoded to have vector representations and cyclical features may need a fourier transform.

5. ML Algorithm Selection: Based on the defined problem a specific algorithm needs to be selected. The type of algorithm can vary whether the problem is a classification or a numerical prediction problem and whether the designer intends a parametric or non parametric approach.

6. Training: Once the algorithm has been selected, the ML model needs to be trained on the feature engineered data. Generally, only 70-80% of the data is used for training. Here the algorithm will take the data and fit it to the model. During this stage, neural networks also perform validation; this will be discussed in the neural network section.

7. Testing: To ensure that the fit of the model to the data is sound, the remaining 20-30% of the data is used to test the model. This testing portion is kept aside so the performance of the model can be assessed with respect to new, unseen data.

3.2 Supervised Learning Approaches

In this study, the following are some of the prediction methods that are used in this study. These prediction methods are all based on supervised learning. The methodology of their implementation will be discussed in the next section. The following are the description of these numerical prediction methods.

3.2.1 Multiple Linear Regression

Multiple linear regression is a modelling method that establishes an equation that approximates the target variable based on the inputs [74]. Multiple linear regression is the multi-variate form of a linear regression and takes on the general form as shown in Equation (1) [74]. Where Y is the dependent variable, all x values are the various, up to n independent variables, the a values are the coefficients that imply a weight between the dependent and independent variable and B a bias value that tends to be an accumulated error for all the variables.

$$Y = a_1 * x_1 + a_2 * x_2 + ... + a_n * x_n + B \tag{1}$$

Some of the considerations in multiple regression is that the dependent variables are to be independent of each other or in practical uses, are not highly correlated and as the name applies, have a linear relationship.

To use this method as a way to predict something, the coefficients and bias are to be determined. The method of solving for these coefficients can be seen as a learning tasks. By taking a data set with both the independent and dependent variables, the coefficients can be either numerically solved or determined by through a learning task such as supervised learning.

However, it is to be noted that real data will not have a perfect linear fit and there will be a spread to the data as seen in Figure 13. The red line in the figure represents the linear fit and as it can be seen, not all the data points lie on the line. Thus, an objective in finding a good fit for any data is to reduce the amount of residuals; or the difference between the predicted and actual value. The mathematical formulation of the total residual can be seen in Equation 2 where the Residual Sum of Squares (RSS) is the sum of all the difference between the true value and the prediction value (value of the fit) [74]. The difference is squared for two reasons; the first to ensure that the residuals underneath the fit and on top do not cancel each other out and for various algorithms it aids in penalizing larger errors.

$$Residuals = RSS = \sum_{a=1}^{m} (y_{actual} - y_{predicted})^2 \tag{2}$$

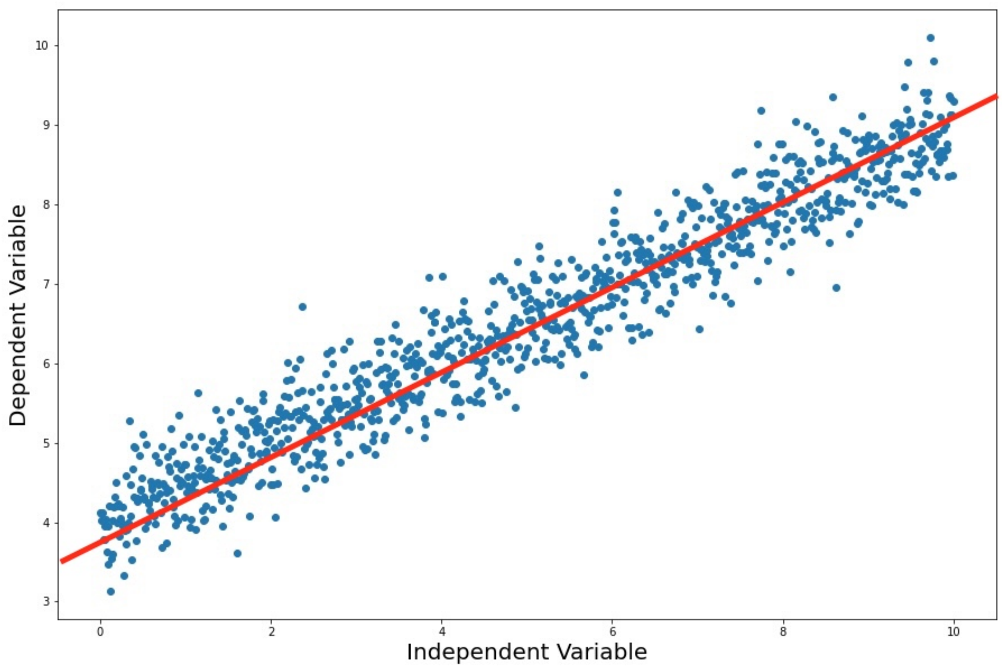

Figure 13: An Example of a linear regression fit on data where there is a spread in the dependent variable.

There are various ways to solve for the coefficients of a multiple regression fit but two popular methods are either through solving the Ordinary Least Squares (OLS) or through learning algorithms such as a neural network.

Ordinary Least Squares (OLS) is a method of reducing the the sum of squares of the residuals through established data. Unlike a linear model, multiple linear regressions have many more variables and to solve for the coefficients, thus a matrix derivation is required as seen in [75].

As seen in Equation 3, the regression fit can be modelled by a Y column matrix of size n which is equivalent to the X input matrix of size n *by* m where m is the number independent variables, matrix multiplied by a column matrix of size m that represents the coefficients s added to the errors ϵ.

$$Y = X * a + \epsilon \tag{3}$$

A better matrix representation of the regression fit can be seen in Equation 4. As the error is not directly measured for each component, the cumulative error is considered as the bias value, B, as seen in the first entry of the coefficient column matrix and the column of 1s in the input matrix. The objective of OLS is to solve for the values in the a matrix given a data set.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1m} \\ 1 & x_{21} & \dots & x_{2m} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_{n1} & \dots & x_{nm} \end{bmatrix} \times \begin{bmatrix} B \\ a_1 \\ \vdots \\ a_m \end{bmatrix} \tag{4}$$

The resultant equation to solve for the column vector can be seen in Equation 5 where X is the input matrix, *X'* is the input matrix transposed and Y is the sample output matrix as derived in [75].

$$a = (X'X)^{-1}X'Y \tag{5}$$

Neural Network Using a neural network to solve for the coefficients of a multiple regression fit uses the same machine learning approach as specified as described later however the only difference is the network design. In packages such as Google's TensorFlow, when designing a multiple linear regression, the network only consists of an input layer with the number of nodes coinciding with the number of input variables, directly fed into an output layer of one node. The package will internally create the number of parameters plus a bias that are to be learnt.

Unlike OLS, the learning model will first arbitrarily set values to the parameters (coefficients and bias) and through the process of Gradient Descent as described in Equation 9, the model will update to a more representative state each learning cycle.

Unlike OLS, instead of using RSS, the Mean-Squared Error (MSE) is used to compare the fitted output and true output as shown in Equation 6.

$$MSE = \frac{1}{n}\sum_{a=1}^{n}(y_{actual} - y_{predicted})^2 \tag{6}$$

A more detailed description of a neural network approach to supervised learning will be discussed in a later section.

3.2.2 Support Vector Regression

Support Vector Machines (SVR) is a machine learning algorithm that predicts a continuous value. It is based on Support Vector Machines (SVM) which is used for classifications. The goal of SVM is to create a line or hyperplane that best separates classes of data into different classifications[76]. When extended to numeric predictions, SVRs use the line or hyperplane as the model fit. Also unlike multiple regressions, SVRs optimize the norm of the coefficient as seen in Equation 7, where a is the coefficient vector, rather than the residuals [76].

$$\mathbf{MIN}\frac{1}{2}||a||^2 \tag{7}$$

An SVR will fit the data to a hyper-plane but also provide bounds to which there is an acceptable error (ϵ). This can be seen in Figure 14 where the red line is the fit of the data and the green lines are the error bounds. This is what differs a multiple regression model to a SVR [76, 77]. SVRs try to fit the data within a certain threshold called the error-tube; the bound between the upper and lower error lines.

The data points outside the tube are used as the support vectors to provide predictions.

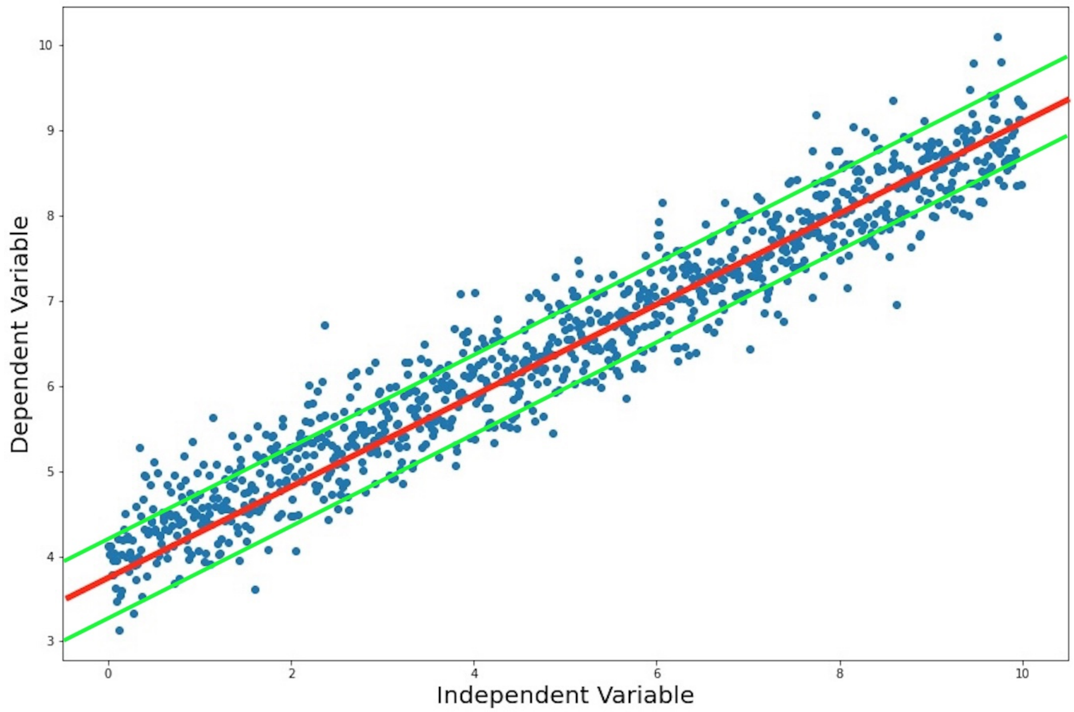

Figure 14: An Example of a Support Vector Regression fit where the red line is the fit and the green lines are the error bounds.

Due to the potential non-linearity of data, creating a fit within the data set's initial dimensionality may not possible. To solve this issue, most SVM algorithms rely on a *kernel trick*. The kernel trick is a mathematical process that takes input and passes it through a *kernel function* that returns the point into a new feature space of higher dimensionality [77]. An example of this can be seen in Figure 15. It is to be noted the kernel example provided is for a classification problem thus the resultant fit is not optimized for a numeric prediction. Some of the most common kernel functions include polynomial kernels with varying degrees representing the increase in feature space or a radial basis function (RBF) that puts the data into a feature space of infinite dimensions as seen in Equation 8 where γ is a hyper-parameter that defines the influence of a single training example [77].

$$RBF = e^{-\gamma * (x-y)^2} \tag{8}$$

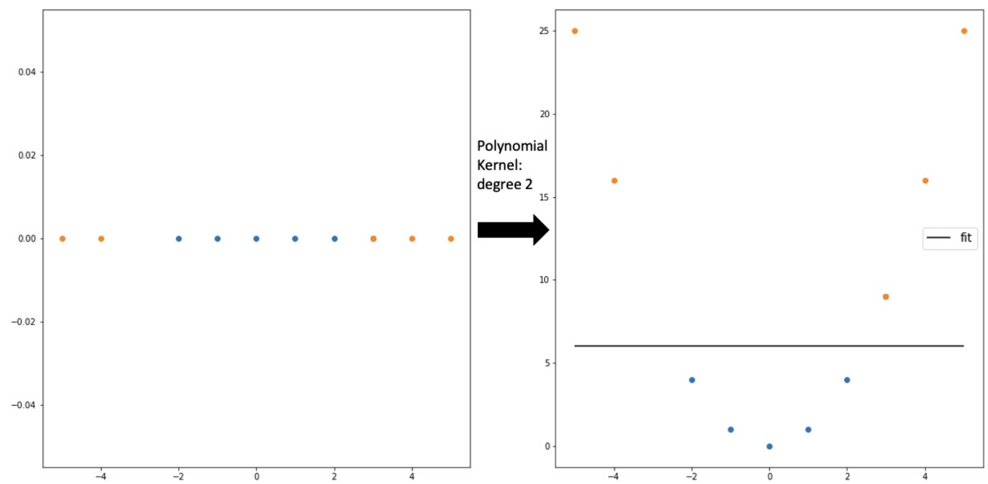

Figure 15: An Example of a polynomial kernel trick where a one-dimensional set of data undergoes a second degree kernel operation to increase the feature space to one where a linear separation fit line can be created.

By setting the error value and kernel functions, the fit can be calculated such as the one implemented by LIBSVM [78]. Another parameter of interest in this algorithm is the value of C, which is another hyper-parameter that controls the allowable error in the SVR fit to prevent over-fitting [77].

3.2.3 Decision Tree Regression

Similar to SVRs, Decision Tree algorithms are heavily used in classification problems, however, they have been extrapolated to predict for continuous values.

The basic idea of a decision tree is to take the data and partition it into smaller branching subsets of similar behaviour. The tree starts with a root node and branches off at various decision nodes that help in the decision making process until it reaches a leaf nodes which are the predictor values. This process is also known as recursive partitioning [79, 80].

The root node of a decision tree is the entire population of the training set. At each each decision node it is a subset of the original data set and nodes where there are no further partitions, the remaining values are averaged in the leaf nodes [79].

To determine the best decision or threshold at a particular node, the algorithm goes through an iterative process by which it calculates some sort of metric to determine the split [79]. Some examples include Gini Impurity for classification problems and Mean-Square Error (MSE) for regression nodes [79]. An example for the MSE use in determining thresholds is seen in Figure 16. It will initialize at a threshold to represent the decision split. It will then determine the averages of the points to the left and the right of the threshold (blue lines). From those respective averages, the MSE is calculated on either side and summated. The threshold is then moved to the next position and the process is repeated. The threshold (red line) is in between two successive points (point1 & point2) and when moved, it is placed between the subsequent point (point2 & point3). After MSE is calculated for all thresholds, the threshold with the lowest MSE is considered the decision for the node.

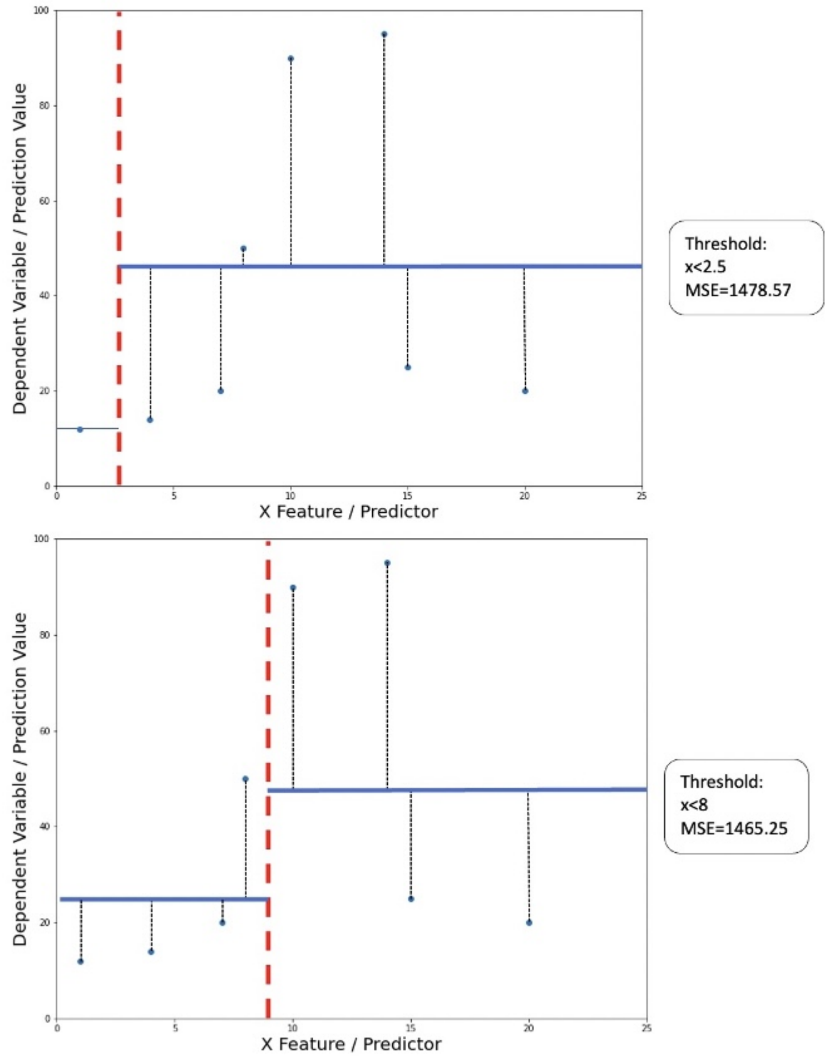

Figure 16: An Example of how thresholds for decision tree nodes are determined by moving the threshold separator and calculating the residuals based on the averages on both sides of the threshold.

However, for data sets with multiple features, the tree not only has to determine the threshold but the best feature to use at a split as well. The general formulation of the decision tree algorithm based on CART implementation is as follows [81]:

1. Start with the original data set and iterate through all the features and calculate the lowest Mean Squared Error and Sum of Residuals as described above.

2. Split data at the node based on the feature with the lowest MSE and set the decision criterion according to the threshold of that MSE.

3. For each of these new subsets, iterate through all the features that have not been previously used as a split criterion and calculate their MSE.

4. Split the subsets into further subsets using the new iterated MSE values.

5. Iterate and perform this process on all new child nodes until features are exhausted.

The aforementioned method describes how decision splits are made but not the formation of the leaf nodes. Once all the features have been iterated through, there is a stop condition on the splits based on a minimum number of samples. Once a subset on either side of a threshold is equal or less than the condition, the remaining points are then averaged and that value is the prediction value [81]. If the condition is not specified, the splits keep on occurring until a single point and this often leads to over fitting.

A sample of a decision tree can be found in Figure 17 where the original data set has 4 attributes. This decision tree has a depth of 3 as the root node is considered to be at a depth of 0. At each of the attributes or features there is a condition and all branches lead to a leaf node which serve to be the prediction nodes.

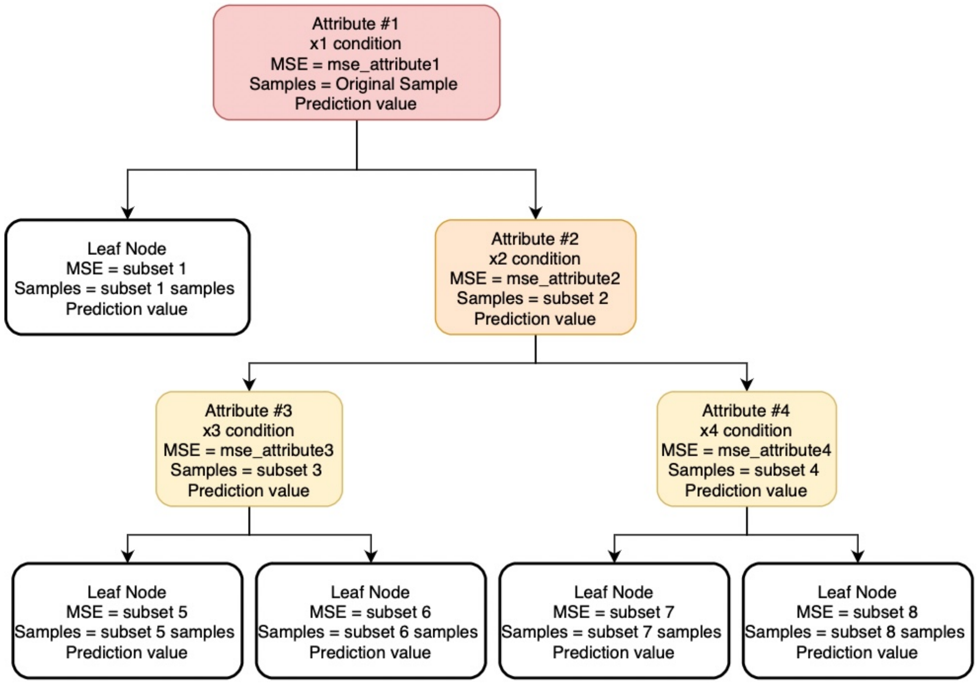

Figure 17: An Example of a decision tree with 4 attributes and the metrics at each node.

One thing to note about decision trees is that they are a non-parametric model meaning their are no internal attributes to be trained. The model is trained based on the creation of the structure of the decision tree. Once the structure is created through feature-label (x,y) pairs, new data can be passed through it. The data will be compared at each decision node based on whether the condition is met will follow the appropriate path. Eventually the data will be led to a leaf node where the value of that node is the prediction for that input.

3.2.4 K Nearest Neighbour

K Nearest Neighbour (KNN) is another non-parametric way to inference a prediction from some input values. In classification problems KNN is used to set up boundaries between clusters of data where each segment is a different category [82].

The KNN regression method plots vertical lines along the feature (x) axis. Then it takes n number of neighbours, or data points closest to the vertical lines based on

Euclidean distance and averages the associated y values [82]. This can be seen in Figure 18 where the 3 red lines represent some sample locations along a feature. The nearest neighbours are located and the averages are computed [82]. Note that Figure 18 is based on a two dimensional example for graphical purposes. The same process would be done in higher feature spaces; hence the use of Euclidean distances.

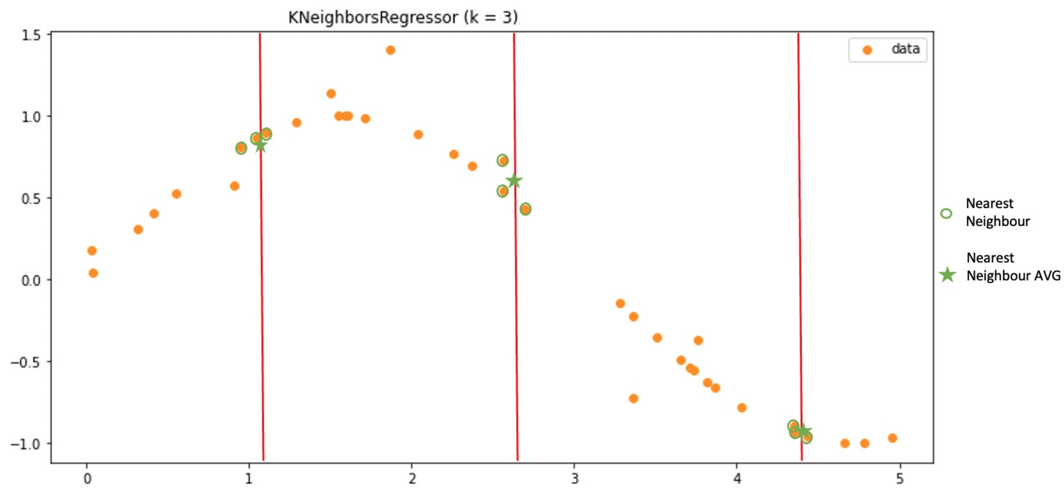

Figure 18: An example of a regression plot where the K nearest neighbours are located in green circles and the averages of the nearest neighbours are the green stars.

This process of locating the nearest neighbours and averaging their targets is done for all points along the feature axis. This results in an interpolation which serves as the algorithm's estimation [82]. This can be seen in Figure 19 where the blue line is the estimation after the KNN algorithm is applied. It is to be noted that the interpolation is sensitive to the value of n as a low n provides a step-wise interpolation whereas a higher n smooths the interpolation [82].

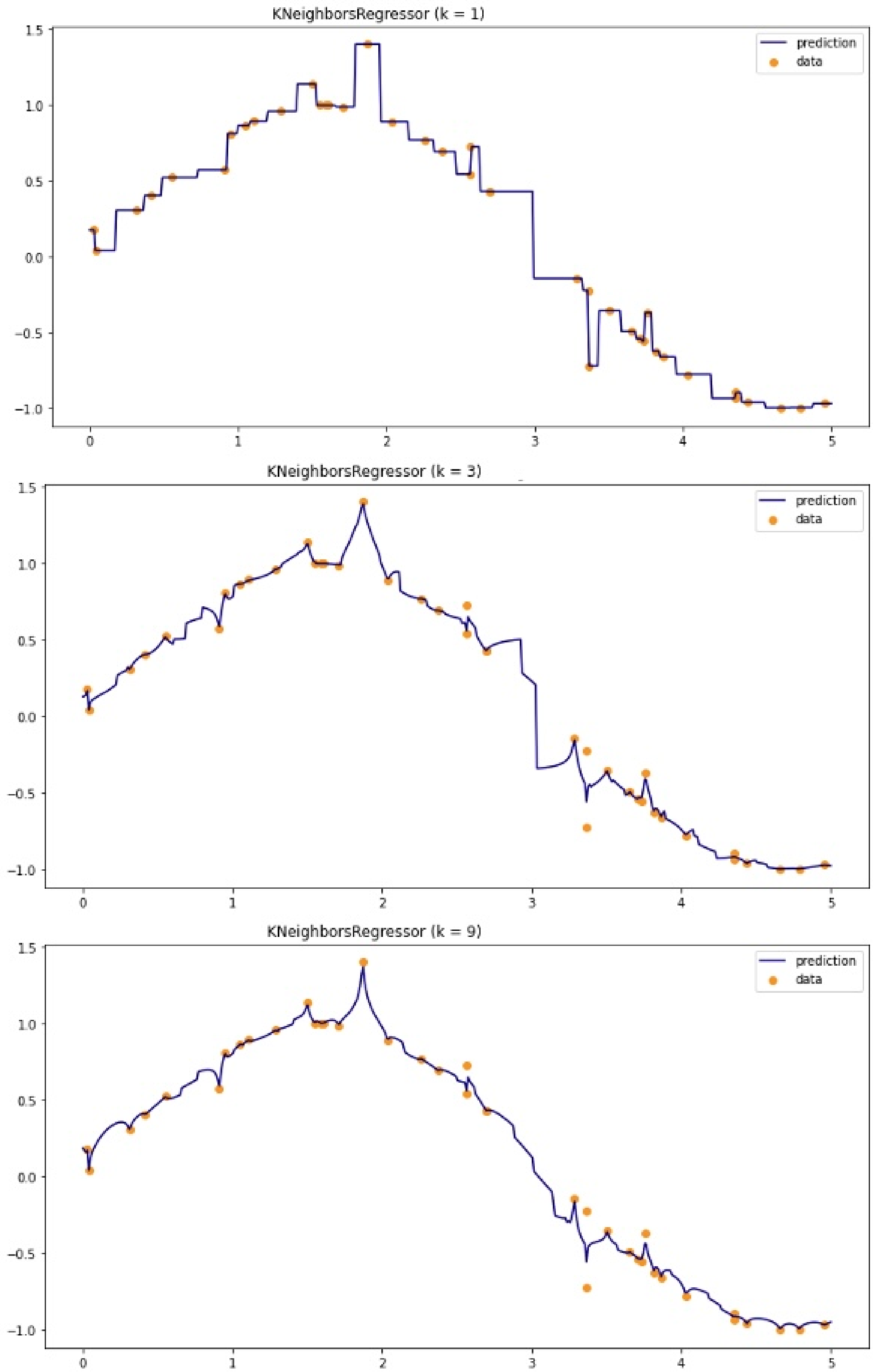

Figure 19: The KNN interpolation of Figure 18 with various values of n.

Variance & Bias: This smoothing behaviour does not mean it is a better fit and this is where the variance & bias tradeoff comes in. Variance is the response to small variations or noise in the training data whereas bias is error between the predicted and expected value.

The tradeoff comes in the fact that both variance and bias are inversely proportional to each other. As the bias decreases, the variance increases meaning the closer the predicted value is to the true value in training, the more sensitive the model will be to noise in data during deployment causing there to be a wider spread in predictions [83].

This relates the K value because the higher it is, the closer the interpolation gets to the actual values. This in tern will increase the variance error and actually lower the accuracy.

3.2.5 Neural Networks

Neural networks are another approach in supervised learning. Here, Artificial Neural Networks (ANN) are created to mimic neurons of a brain. A network of nodes and connections are made to map the input to an output.

Creating a machine learning model is usually broken up into two different stages. The first one is training where the parameters or the *tuning knobs* of the model are being learnt through repeated exposure to data. The model then compares a predicted output to the true output to determine the change needed in the tuning knobs to be a more representative of the data. The second stage is a testing stage where a subset of the data that the model does not see during training is passed through the trained model to compare the true and and predicted outputs to test the accuracy.

The general pipeline for a training an ANN can be seen in Figure 20 and each component is discussed in the following sections based on [65] and [66].

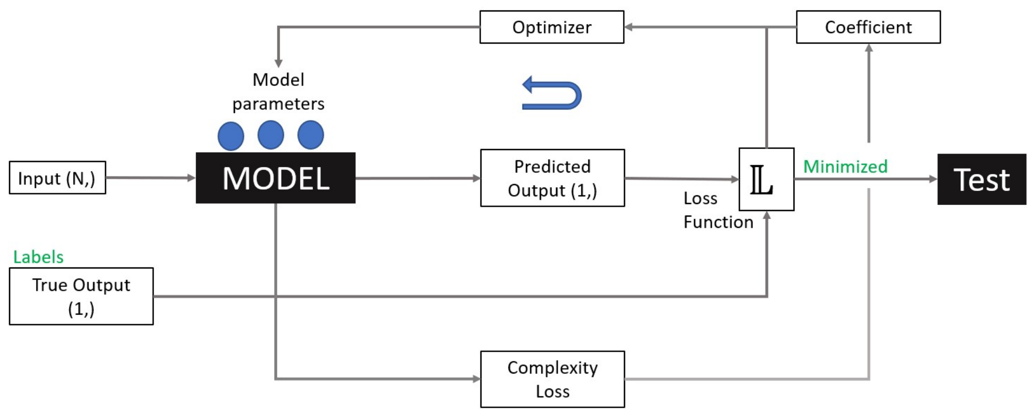

Figure 20: The learning pipeline for a Supervised Learning Task.

Input & True Output The input is a vector of N features that have undergone feature engineering which correspond to the row of the data set. ANN packages such as Tensorflow use parallel computing thus process multiple rows at a time; typical number being 16.

The true output are the labels (dependent values) which are used to tune the parameters of the ANN to.

Neural Network: The neural network or other network types are the internal structure of the algorithm that is trying to be learnt through the the iterative process of prediction and optimization. There are different types of models that can be used such as Recurrent Neural Networks (RNN), Deep Neural Networks (DNN), Convolutional Neural Networks (CNN) [65, 66]. For this study DNNs are used and will be discussed in a later section. Each of these models has model parameters which are the components mapping the relationships between the input and output. In a linear model, the model parameters are the slope of the line and the intercept; often generalized as the bias. A sample neural network can be seen in Figure 21. The input layer takes the input vector with each attribute having its own input node, passes it through hidden layers that transform the values using weights and bias values to then be passed though the output layer that provides the prediction [65].

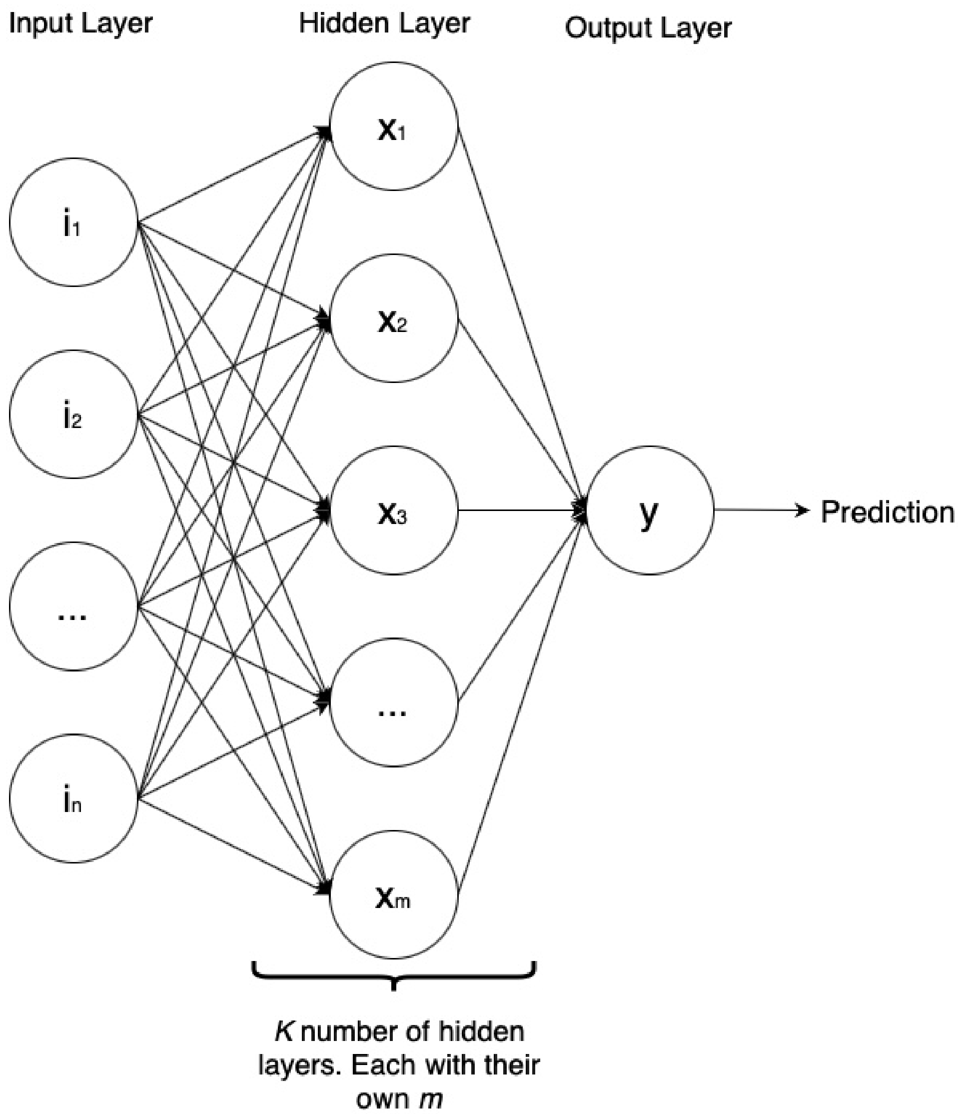

Figure 21: An Example of a Neural Network.

Predicted Output: In each training iteration, the inputs are passed through the model and given the current state of the model parameters, a predicted output is generated.

Loss Function: A loss function is a mathematical expression that takes the predicted output and the true output and compares their similarity. The more similar the prediction and the true output are, the lower the value of the loss function [65, 66]. Depending on the type of network used and whether the learning task is a regression or classification, different types of loss functions are used. Some of the more common loss functions are Mean-Squared-Error (MSE), Cross-Entropy, and Mean-Absolute-Error (MAE) [65, 66]. The reduction in the loss function is the objective of the learning task.

Complexity Loss & Coefficient: Complexity Loss is an optional component to the learning phase and is part of the parameter regularization. It is method for fixing over fitting. This will be discussed more in later sections [65, 66]. Generally, if a model is overfit, the parameters and their values have too much of an influence on the data. This complexity loss penalizes the usage of unnecessary model parameters. The coefficient in Figure 20 reduces the complexity loss's influence on the optimizer such that the Loss Function is still the primary factor for optimization.

Optimizer: The optimizer takes the current value of a model parameter and the loss function of a given epoch (learning cycle) and updates the model parameters. This optimization is done based by differentiating the Loss function with respect to the model parameters as the gradient of the loss function represents the rate at which a model parameter is reaching a better representative state [65, 66]. One of the most popular optimization methods is Gradient Descent. The mathematical formulation can be seen in Equation (9) where θ_{i-1} is the current value of a model parameter in the current epoch [65, 66]. θ_i is the new calculated model parameter value to be used in the next epoch. $\nabla\mathbb{L}$ is the gradient of the loss function calculated and α is the learning rate. The learning rate is a very small number that reduces the size the gradient is changed by. This is to ensure that optimization reaches a local minima in small iterative steps and the model parameter does not diverge do to overshooting. The closer $\nabla \mathbb{L}$ is to 0, the closer the model parameters reaching to a trained state.

$$\theta_i = \theta_{i-1} - \alpha * \nabla \mathbb{L} \tag{9}$$

Test: As mentioned, only 70 to 80% of the data set is used for training, the remaining is used for testing [65, 66]. The remaining data set is not passed through the training phase and is used to test the performance of the model before deployment. Generally, the test data has variations to test overfitting and also serves as an indication if the true accuracy of the model is sufficient. All the data is passed through the model and the predicted output is compared to the labeled output of the test set to determine the overall accuracy [65, 66]. This is done because the Loss function is not an indication to the performance of the model, only an indication of the learning process.

3.2.6 Network Design

In machine learning, the the neural network is the structure or architecture on which the model is trained [65]. The design of the model directly affects the complexity, computation requirements and ultimately the accuracy of the trained model such as the one in Figure 21.

The nodes in each layer represent the neurons of the model and increasing the number of nodes in each layer increases the number of computational units. The edges connecting the nodes to each other are the weights which are updated during learning. Once the network is trained, the input passes through the various nodes and edges according to their weights.

In ML, a common term used is *Hyperparameters.* Hyperparameters are a predetermined set of model parameters or "tuning knobs" dependant on the network architecture. The values of these parameters, such as the weights, are updated through the learning process and represent the hidden statistical mapping from input to out put. The number of model parameters in a simple Feed-Forward Neural

Network can be represented by the number of network edges plus bias for each node after the input layer [65].

The following are a description of some neural network layers in neural networks.

Input Layer: Has a number of nodes corresponding to how many input features there are in the data set. The network can only have an input node size dependant on how many features are available. Null nodes cannot be added.

Hidden Layers: The number of layers and the node size of each layer is user set and subsequently affects the number of edges. These hidden layers perform non-linear transformations on the input data based on trained weights and biases [65, 66]. There are different types of hidden layers that can be used depending on the application. However, for this study, dense hidden layers are used. Dense layers are layers that deeply connect inputs to intermediate nodes.

Figure 22 is a sample of information flow in a neural network. In the figure, i are the inputs, w are the weights and x is the intermediate value at a hidden layer node. σ is an activation function that is used to normalize the data and prevent extrema from biasing the calculations. At every node, there is a b bias value as well.

Equation 10, presents how the inputs are transformed at each node of a hidden dense layer. All the inputs are multiplied with their respective weight and summated with each other and the bias value [65, 66]. As noted the weights and bias are what the network is trying to train, thus at every epoch, the weights and bias have different values.

$$x = \sum_{a=1}^{n} i_a * w_a + b \tag{10}$$

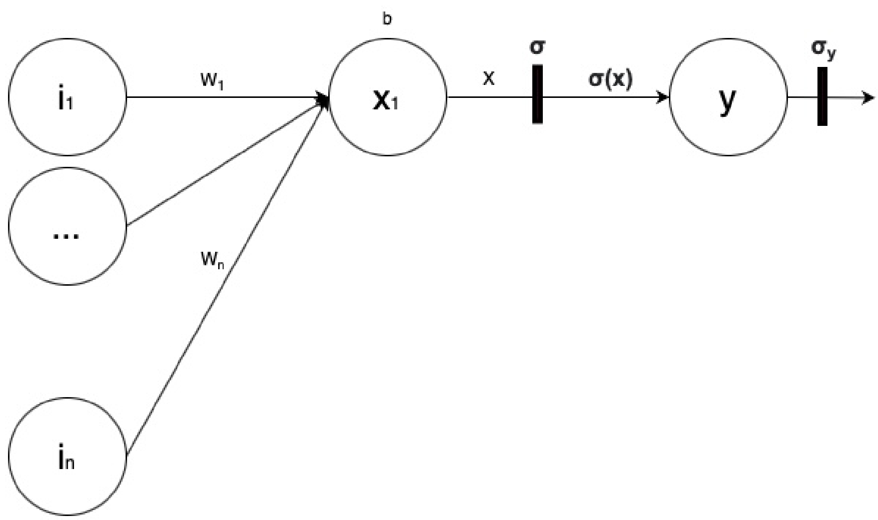

Figure 22: An Example of the flow of inputs to an individual node in a hidden layer where i are the inputs, w are the weights and x is the intermediate value at a hidden layer node.

This process is applied for each node in the hidden layer and if there are multiple hidden layers.

Activation Layer: There are various types of activation layers that are user set however the size is not. These are numerical transformations and the size is dependant on the layer before it [65, 66]. Neural network weights can cause the input values to blow up and these activation functions normalize the values to certain ranges to prevent extrema behaviour. These functions cannot change the shape of a dense layer output. For example, a common activation function is the *sigmoid* function that transforms its input into a binary range of 0 or 1 as seen in Equation 11 or the *relu* (rectified linear) that transforms negative numbers to 0 and the rest to a linear range as seen in Equation 12 [65].

$$\sigma = \frac{1}{(1 + e^{-x}} \tag{11}$$

$$\sigma(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \tag{12}$$

Output Layer: These layers are dependant on the type of model chosen. Classification models have nodes corresponding to the number of categories and numeric prediction models have a single output node such as the one shown in Figure 23. The output of the neural network is the prediction that will then be compared to the labeled value.

As seen in the figure, all the intermediate values at the hidden dense layer are passed through an activation function and those values, $\sigma(b)$, are summated with a final bias b_y as seen in Equation 13. The final prediction can also go through a transformation function such a sigmoid for binary classifications or, as seen in 14, a linear transformation.

$$y = \sum_{a=1}^{m} \sigma(b) + b_y \tag{13}$$

$$Prediction = \sigma(y) = y \tag{14}$$

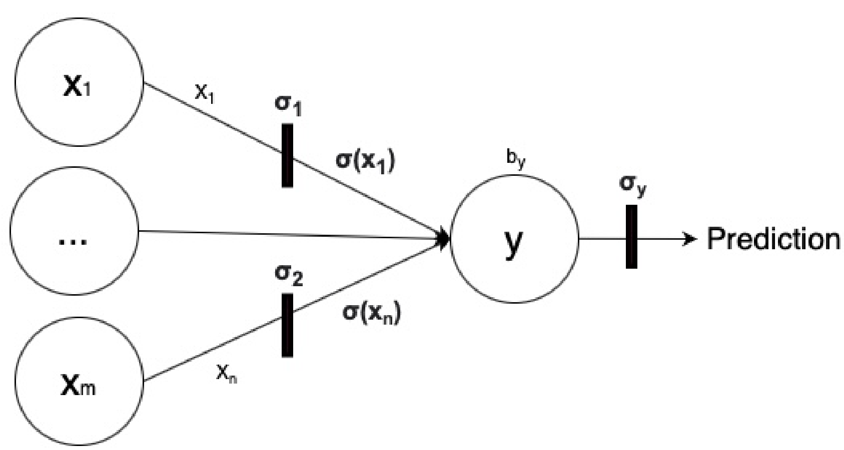

Figure 23: An Example of the flow of values from hidden and activation layers to the output layer in a Artificial Neural Network.

Validation and Over Fitting Validation is the testing step as mentioned earlier. In neural network models, to assess the model during training instead of having to wait until the testing phase can help catch errors in training much earlier [65, 66]. Validation is performed by taking a subset of the training data and setting it aside. During each epoch, as the loss is calculated so is the accuracy of that validation set. For every epoch, a different set of values from the training is used for validation. Thought, during training the network has seen the validation data, it is still representative enough to determine if over fitting is occurring. This is different than the training accuracy as, every epoch the network will also assess the accuracy in predicting the data it just trained whereas validation is kept to away [65, 66].

There are multiple reasons as to why the validation accuracy can be lower than the training accuracy. The first possibility is that the validation data set has instances which the model is not trained on. This is when there are samples where the input values are in a combination or ranges that the model did not experience or account for in training or there are new output values where there was insufficient training [65, 66]. An example of this can be seen with quality classification models where higher quality instances are less likely to be seen.

However, this is a result of a poor testing and training split. It is important to make sure that both the training and validation sets have roughly the same percent of output instances. For example, in a numeric prediction model such as trying to predict the temperature of a system, the training and validation should have the same percentage of instances for both low temperatures and higher temperatures. This provides variety for the model to be trained on and a more appropriate data for testing.

3.2.7 Principal Component Analysis

One common practice in data analysis is to reduce the dimensionality the data in a meaningful way. A reduction in dimensionality aids in preserving computational resources however, removing the wrong features in a data set may affect the model in a negative way with respect to accuracy, thus a systematic approach is required.

Principal Component Analysis (PCA) is a statistical technique that reduces dimenstionality while trying to reduce information loss [84].

The general approach to PCA is as follows [85]:

1. Separate the dependent and independent variables as PCA is only applied to the dependent variables.
2. Calculate the mean and standard deviation for each column.
3. For each column, normalize the data by subtracting the mean and dividing by the standard deviation of that column.
4. Calculate the covariance matrix of the features in the data set.
5. Perform eigen decompositon on the covarience matrix to obtain the eigenvalue and eigenvectors of each feature.
6. Order the eigenvalues from greatest value to least and then shuffle the eigenvectors such that the features correspond to the ordering of the eigenvalues. Note that the eigenvalues represent the variance each principal component has on the data.
7. Determine a variance threshold which represents how much oh the total variance a reduced data set is to have. This threshold can be pre-determined or set based on results of a scree plot as seen in Figure 24.

Scree Plots plot the variance of each principal component in an additive way. This means the first principal component has the highest contribution to the total variance and subsequent principal components have lower contributions (hence the ordering in step 6). A principal component's variance is the difference between it and the previous principal component. This plot is useful as it provides a graphical representation on when adding additional principal components provides diminishing returns on the total variance. As seen in Figure 24, at principal component 7, the addition of more does not add a significant amount to the total variance but their inclusion may significantly impact the amount of computational resources. As such, all principal components past 7 are removed.

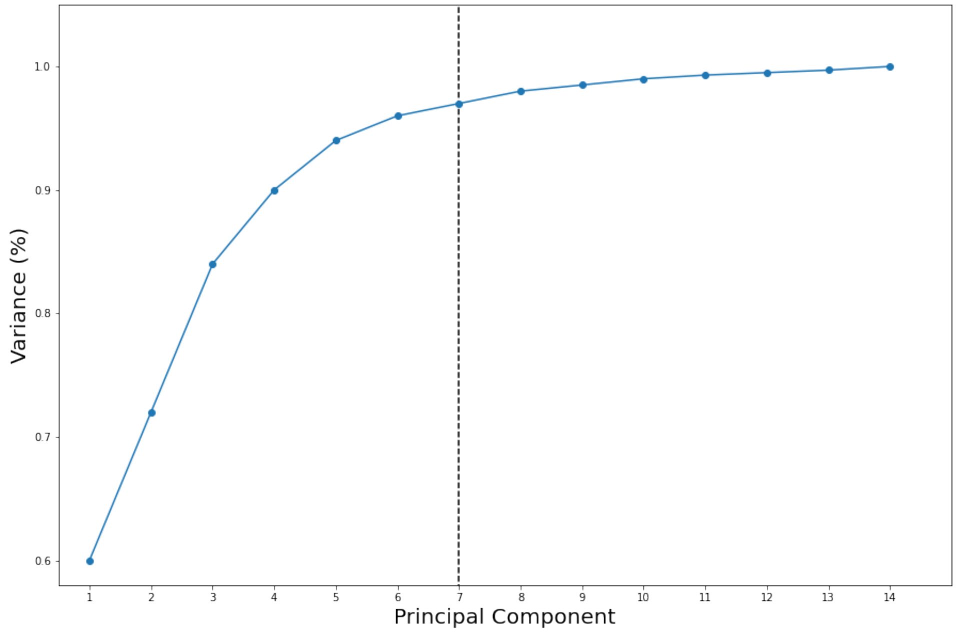

Figure 24: An example of a scree plot where all principal components and their variance contributions are plotted. The black line represents a variance threshold used to reduce the data set.

4 Methods and Approach

The work first presented information on maintenance practices in nuclear power plants and how delayed maintenance tasks can lead to deferred maintenance. Then the work presented information on Machine Learning practices, in particular, discussion of various supervised learning algorithms. Both in the aforementioned literature review.

This section outlines the methodology on how a proof of concept was created to test and understand the implementation of the various algorithms for the purposes of predicting maintenance delays.

4.1 Data Set Selection

Obtaining operational maintenance data for nuclear power plants in the public domain is a difficult task as it provides context on a company's performance with respect to maintenance costs and is often kept as proprietary information.

The objective of this work is to be able to apply various data techniques to predict delays. Considering that and the data availability restrictions as mentioned in the literature review, a representative data set was chosen to complete this work.

The data selected for this work was the *Condition Based Maintenance of Naval Propulsion Plants* data set from the University of California Irvine Machine Learning Repository and its accompanied work by Cipollini et al, on *Condition Based Maintenance of Naval Propulsion Systems with Supervised Data Analysis* [86].

The data was created based on a "real-data validated complex numerical simulator of a Navy frigate" [86]. It presents data that is characterized by a *COmbined Diesel ELectric And Gas* (CODLAG) propulsion plant as seen in Figure 25. Figure 26 presents the breakdown of the gas turbine which this data set represents

The fuel as well as the compressed air from the compressor drive a set of turbines as seen in Figure 26. This gas-turbine system then undergoes a set of gear reductions to separate the power into two streams. The rotational power from the two streams powers an electric engine which drives the propeller and creates movement for the ship.

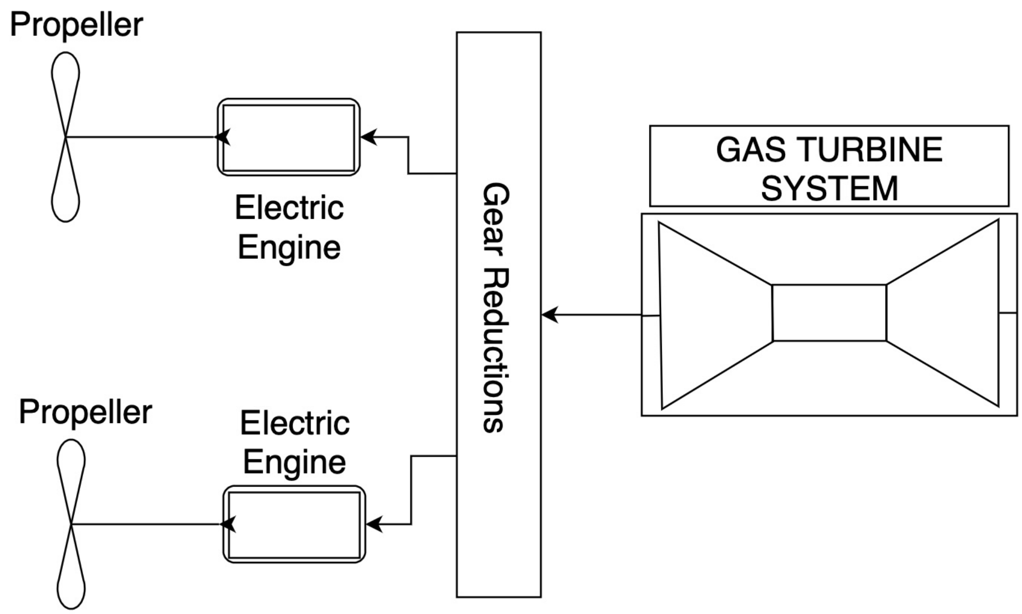

Figure 25: Schematic for a COmbined Diesel ELectric And Gas (CODLAG) propulsion plant

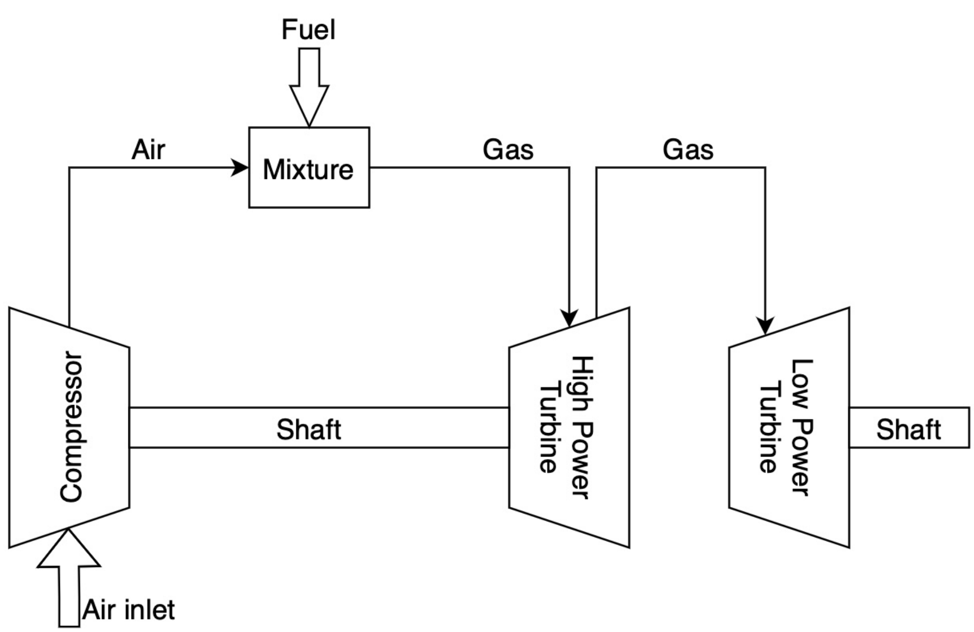

Figure 26: Schematic for the Gas Turbine System in the CODLAG propulsion plant

The following table are the list of the attributes in the data set. As noted by [87], the only attribute that is controlled by user input is the lever position (attribute 1) otherwise these are sensor readings direct from the components. As it can be seen, the components all play a role in the overall function and if any of the components in the turbine generating system were to not function, the propulsion capabilities would seize. This exact turbine system and behaviour is also seen in nuclear power plants.

Attributes 1 to 16 represent variables of the propulsion system where the measurements are taken from sensors. These measurements are often standard measurements taken in most naval propulsion plants and many are also taken in gas and nuclear plants such as turbine inlet and outlet temperatures.

Attribute 17 and 18 are the labels and the attributes trying to be predicted. Gas turbine and the gas turbine compressor decay state coefficients are unit less metrics that range from 1 to 0 and represent the performance of their respected components. In this application, the performance is based on the reduction of air flow rate through the engine over service hours [88, 89]. The higher the decay coefficient, the more the particular component is decayed and requires maintenance. The same components and systems can be found in a nuclear power plants thus making this a good representative data set. Some of the differences include *Lever position* and *Ship speed* as those are specific to a naval ship, however, it is postulated that information such as reactor power can be used as a replacement.

This data was used as it is simulator data and is clean for analysis. This allows for sensitivity analysis which will be discussed later in the outline of experiments. Also, the type of data is in line with the type of data that can be found in nuclear power plants.

Though this is a good representative data set, there are a few limitations. First, the data set only samples end of decay cycle values; 0.95 and 0.975 to 1 for the compressor and turbine decay respectively though the simulator ran complete cycles from 0 to 1. This is not expected to affect the prediction ability significantly under the assumption the decay trend is linear throughout just like the data set values. Another limitation is the lack of time scale on the sampling points in the simulator. The source of the data has not provided a time scale on which the decay occurs on so it is unknown whether it is over a span of months or years. This however does not affect the prediction ability as there are no time attribute. The time is only useful in understanding the behaviour of the system. As seen later in the pre-processing sections, the instances are all shuffled so the model has no context on temporal affects in the system.

There are a total 11934 samples and is labeled as a numeric prediction task.

Table 2: Naval Propulsion Maintenance data set Attributes

#	Attribute	Units
1	Lever position	[#]
2	Ship speed	[knots]
3	Gas Turbine shaft torque	[kN m]
4	Gas Turbine rate of revolutions	[rpm]
5	Gas Generator rate of revolutions	[rpm]
6	Starboard Propeller Torque	[kN]
7	Port Propeller Torque	[kN]
8	HP Turbine exit temperature	[C]
9	Gas Turbine Compressor inlet air temperature	[C]
10	Gas Turbine Compressor outlet air temperature	[C]
11	HP Turbine exit pressure	[bar]
12	Gas Turbine Compressor inlet air pressure	[bar]
13	Gas Turbine Compressor outlet air pressure	[bar]
14	Gas Turbine exhaust gas pressure	[bar]
15	Turbine Injecton Control	[%]
16	Fuel flow	[kg/s]
17	Gas Turbine Compressor decay state coefficient	[unit less]
18	Gas Turbine Turbine decay state coefficient	[unit less]

4.2 Reproducibility

In order to make this work reproducible, the following set up was used to conduct all experiments.

- Programming Language: Python 3.7.11
- Machine Learning Package: Tensorflow 2.60 & scikit-learn (sklearn) 0.22.2.post1
- Operating Environment: Google Colab with GPU acceleration
- Memory: 25GB GB of RAM
- Processor: Intel(R) Xeon(R) CPU @ 2.30GHz (4 Cores)
- numpy version: 1.19.5
- pandas version: 1.1.5

4.3 Preliminary Data Analysis

The first step in this work was to have a better understanding of the data and explore any behaviours that may alter the selection of algorithm or set up requirements for implementation. In order to do this, some preliminary data analysis was performed.

The following were the steps that were used to perform the preliminary data analysis on the data and what was the objective of completing each step of each step.

4.3.1 Variable Exploration

The first step in the preliminary data analysis is to understand the variables individually. This includes determining their basic statistical information including; mean, standard deviation, skewness, min & max values, and the 25%, 50% and 75% percentiles. Other additional metrics to consider for each category is counts and number of unique values.

The statistical analysis will provide a better understanding on the features and can be compared to the theory. Also, understanding the data can aid in the data manipulation. This step is also useful as it can dictate what types of algorithms can be used on the data. For example, if the attributes have a high degree of skewness, statistical models assuming normal distributions cannot work.

In addition to the statistical descriptions of the variables, some visualization was completed in order to get a better representation of said description. For each variable, a boxplot was created to visualize how the data is distributed and also get a visual representation of any potential outliers that may have to be addressed in data cleaning. Additionally, a probability density histogram was created to understand the spread of the data.

4.3.2 Data Cleaning

The results from the previous step will aid in cleaning up the data. Some of the first steps is to determine how many rows in the data set have missing entries and how many duplicate entries there are. For both instances pandas internal functions were used and none were to be found.

Based on unique counts, both Compressor Inlet Air Temperature and Pressure attributes were removed as they only had one unique value each.

Then, according to the box plots, outliers were visualized for each of the variables to determine whether they affect the prediction ability in any significant way. The turbine injection control was the only variable with outliers however, these were kept in the data set. It is to be noted that there was not a need for vast data cleaning as this is simulator data. More will be discussed in the results section.

4.3.3 Relationship Analysis

Once information on the individual attributes were determined, it was important to determine the relationship between them. This aids in determining if there are any attributes that provide no influence on the target attribute and can be removed. In this study, this analysis was done by having a correlation heatmap and pairplots.

The correlation heatmap plots the *Pearson Correlation Coefficient (R)* of each attribute plotted to each other in a matrix format. The heatmap for this work is presented in Section 5.1.3 The range of the correlations are from -1 to 1; where the higher the magnitude of the number, the more correlated they are, and the sign representing whether they are directly or inversely proportional to each other, positive meaning directly. This aids in determining the relation of the variables. The model designer can also determine if the removal of certain attributes will harm or benefit the prediction abilities. For example, if an attribute has minimal to no correlation to the other variables, its removal can serve to benefit the prediction as the algorithms try not to optimize for that.

Pairplots were also created by plotting all the values of each attribute to each other. In this case, a 18 by 18 matrix of pairplots was created with each position plotting two of the variables against each other. This provides a visual representation of the correlation factors from the heatmap.

On the diagonals of the heatmap, all the values are 1 as they are correlating the variables to themselves. In the pairplot the graphs are plotting attributes to themselves on the diagonals, so the package automatically plots the histrogram. For both plots the top half on the diagonal is inverse to the graphs on the bottom diagonal and in the heatmap they are the same value.

4.4 Algorithms

The following section outlines the different algorithms that were selected for this study and how they were implemented.

The algorithm's selected in this work can be grouped into two different categories. The first being the *SKlearn* algorithms which are based on the *SKlearn* machine learning python package and the second being the *Tensorflow* neural network algorithms which are based on Google's *Tensorflow* machine learning python package.

The two separate packages were used in this study for various reasons. *Sklearn* is a high level machine learning package with many different algorithms pre-programmed and are callable as objects in python. This allows for rapid implementation of various algorithms and is an excellent tool for prototyping or quick analysis. This package is also used for the data pre-processing functionality. However, Tensorflow is machine learning package that is specialized in deep learning, in particular with neural networks and differentiable programming. There are many different types of deep learning packages similar to Tensorflow such as *Pytorch*. However, Tensorflow was used due to its extensive use in various industries, its operability with Google Colab, the programming environment for this work, and its implementation with *Keras*, a python library that provides a user friendly approach to design and implement neural networks.

4.4.1 Data Preparation

The data for all these algorithms had to be divided based on input vectors, labels and also split based on training and testing. The splitting for the input and labels was done using *numpy* and the splitting into training and testing was done using *sklearn train_test_split*.

As this data had two possible prediction values, Gas Turbine Turbine Decay and Gas Turbine Compressor Decay, there are 2 separate labels, one for each separate prediction attribute. Each target variables is considered its own problem. The remaining 14 attributes were separated into their own input array.

The train-test split was done using a 80-20% ratio respectively. Resulting in 9547 training instances and 2387 testing instances. The split was completed was completed using a shuffle feature to randomly move instances around so they are not split based on the sequence in which they were recorded. The random state for the shuffle used a seed of 14 for the sake of reproducibility.

A separate set of input variables were created that standardized the original values using the *SKlearn StandardScaler* implementations. The standardization was completed as seen in Equation 15, where μ is the mean of the feature and σ is the standard deviation of the feature. This process was done for all feature columns so that the features can be centered around a mean of 0 a standard deviation of 1. This feature scaling ensures that attributes of different ranges do not influence various algorithms differently.

$$x_{i-standardized} = \frac{x_i - \mu}{\sigma} \tag{15}$$

4.4.2 SKlearn Implementations

This section outlines the implementations of the algorithms in aforementioned literature review based on the *scikit learn* machine learning package.

The following SKlearn algorithms were selected to provide a variety in this study for various non-deep-learning algorithms. Parametric models such as support vector regression and multiple regression were selected as well as non parametric models such as the decision tree and k-nearest-neighbour. Support vector regression was selected do to its use in existing work with similar data as described in [87]. Multiple regression was selected due to its extensive use in industry for other applications and would provide a comparison to other algorithms to something that is widely adopted. Decision trees were selected due to its ability to generate a visual representation which can be helpful when trying to understand how the algorithm came to its prediction. The K-nearest neighbour algorithm was it is often used due to its simplicity in implementing and interpretation.

Support Vector Regression The two parameters of interest for the SVR algorithm were the Regularization parameter (C) that regulates the trade off between the bias and variance when trying to create a decision boundary and (ϵ) which dictates the allowable error tolerance. The work tested which combination of these two parameters harboured desired results. To perform the support vector regression, a *radial basis function* (rbf) kernel was used. The default values of the package for the error ϵ-tube (0.1) and for the *l2* regularization penalty (1) was used.

Multiple Regression Sklearn approaches multiple linear regressions using Ordinary Least Squares (OLS). The default parameters were used, including solving for a bias intercept value.

K-Nearest Neighbour In this implementation, the number of nearest neighbours was *n=5* and all had uniform weights when averaging. Sklearn's algorithm to iterate over the feature space was auto-selected internally among either, *ball_tree, kd_tree* or *brute-force search*. Euclidean distance was used to determine distance of points. Another set of experiments were ran using *n=2, 3, 7 & 10*. In the SKlearn implementation of the KNN Regressor, the default neighbour value is set at 5. To determine the sensitivity to the numbr of neighbours, this value was both increased and decreased. In both instances, when a decrease in prediction ability was seen, the experiment was stopped.

Decision Tree Regressor The criterion for selecting the best node was based on the mean-squared-error (MSE) with the condition of *best* instead of random. No limit was placed on the depth of the tree and a minimum samples for split requirement was set at 2 samples. Another set of experiments were ran testing the algorithm using 5, 10 and 50 minimum samples for a split.

4.4.3 Tensorflow Implementation

This section outlines the implementations of the algorithms in the aforementioned literature review based on the *Tensorflow* machine learning package.

Deep learning and neural networks were selected in this work to study if these advanced approaches to data analysis can help modernize nuclear power plants. SKlearn algorithms are more traditional, statistically driven approaches to supervised learning but are limited in the scale. As mentioned in the literature review, deep learning and big data allowed for more scaling of data. As traditional algorithms hit a plateau of performance with more data, deep learning algorithms can be improved to perform better with more training data. This is particularly important to understand as nuclear power plants have thousands of maintenance tasks with an influx of new emerging tasks, and being able to scale up with more data is pivatol in creating a viable solution. Hence, deep learning was selected.

The loss function used for these algorithms were the *MSE* function and the ADAM optimizer was used with various runs of an initial value of 1, 0.1 and 0.001. Training cycles (epochs) tested were 10, 20, 50, 300, 1000 and 2000. A validation split of 20% was used to estimate the testing results during training. The use of MSE for the loss function is based off similar work done with this data set as seen in [90] & [91]. ADAM was used as the optimizer of choice to implement an adaptive learning rate in order to save on training time.

Multiple Regression Network As this is a multiple linear regression model, the network consisted of an input layer of 14 corresponding to the number of input features directly fed into a dense output layer of 1 with a corresponding linear activation function.

Artificial Neural Network The neural network design was a feed-forward network consisting of an input layer of 14 to a dense layer of 14 nodes to another dense layer of 7 and a final dense output layer of 1 node. For the 2 intermediate dense layers, a *relu* activation function was used and a linear output activation. Other variations of the network were tested with only one intermediate dense layer of 7 nodes and one with 3 intermediate layers of 21, 14 & 7 nodes respectively.

4.5 Experimental Runs

The following are the different experimental runs that were performed. Their results and discussion will be presented in the following sections.

4.5.1 Inital Runs

To establish baseline results, all the algorithms are ran to predict the compressor decay coefficients. For this set of experiments, the following particular configurations were used certain algorithms:

- Support Vector Regression: $C = 1$ & $\epsilon = 0.1$.
- K-Nearest Neighbour: $n = 5$
- Decision Tree: Minimum samples for split is 2.
- Neural Networks: ADAM optimizer = 0.001 and epochs=10, Network shape of 14(input), 14(dense), 7(dense), 1(dense output).

4.5.2 Sensitivity of Various Configurations

The following runs were done to test the difference occurred by changing the configurations for particular algorithms

- Support Vector Regression: testing regularization parameter values of C= 0.5, 0.75, 1, 2 and 4 & the allowable error ϵ=0.1, 0.01 & 0.001.
- K-Nearest Neighbour: testing n= 2 and n= 10.
- Decision Tree: Increasing minimum samples for split to 5, 10 & 50 samples.
- Neural Networks: ADAM optimizer = 0.1 and 1.
- Neural Networks: Increase epochs to 20, 50, 200 and some at 1000.
- Neural Networks: Test the 2 other proposed network designs.

4.5.3 Sensitivity of Prediction Ability with Addition of Noise

In these sets of experiments, to simulate how prediction ability would be affected by predicting the decay using noisy data but clean data for training, the following was simulated:

- Training the algorithms with clean data but testing it with noisy test data. Noise to the testing set of the data was added uniformly across all values with the ranges of noise $\pm 0.1\%, \pm 1\%, \pm 2.5\%$ and $\pm 5\%$.
- Training and testing sets both incurred noise uniformly distributed to the entirety of the set with ranges of $\pm 0.1\%, \pm 1\%, \pm 2\%$ and $\pm 5\%$.
- Training and testing sets both incurred noise uniformly distributed with ranges of $\pm 1\%, \pm 5\%$ and $\pm 10\%$. However, they were not distributed to the entirety of the data sets rather selectively distributed. Tests were done with distributing the error to 10% and 25% of the training and testing sets. A seed of 14 was used to randomly select the affected data.

A sample of how the noise changed the input data can be seen in Figure 27. The addition of noise was done in ranges to add some randomness to the input data and percents were used due to the different scaling of each of the features.

Original Data	Uniform Noise	Selective noise
10	10.5	10
20	19.75	20
15	14.67	15
10	10.24	10
20	20.5	20.5
30	29.76	30
45	44	44
50	52.5	50
10	10.25	10
35	35.05	35

Noise range of [-5%,5%]
with uniform distribution
On entire dataset

Noise range of [-5%,5%] with
uniform distribution
On random 20% of the data set

Figure 27: An example of how noise was added for the various sensitivity experiments.

4.5.4 Sensitivity of Prediction Ability with Dimensionality Reduction using Principal Component Analysis

To test the sensitivity of the prediction ability of the various algorithms with reduced dimensionality, Principal component analysis was conducted to reduce the size of the input data. As such, the data underwent a PCA reduction and was ran using the best configurations of the algorithms from the previous steps. The number of principal components selected were 1,2,4 & 7.

4.5.5 Metrics

For the *SKlearn* models, the package uses R^2 value as a score with 1 being the best possible score. This metric is suitable for the application because it provides a standardized metric that can be used to compared the fit of the various algorithms to each other.

However, other metrics were used to contextualize the prediction abilities rather than just the prediction trend. The mean squared error (MSE) and the mean absolute error (MAE) were used as an indication of how much potential error the prediction will have. The implications of this will be addressed in the results and discussion section.

However, *Tensorflow* does not have a R^2 metric so the MSE and the MAE was calculated for all the models and used as the primary comparision metrics. This will be useful as it will provide an average value of how much error each model will provide that is comparable to the results in the *SKlearn* implementations. The MAE provides context on the extent of the error and the MSE provides a gauge on how spread the errors and weather they are consistent through all the testing points or not. Similar work using neural networks and the same data set also used MSE and MAE as performance metrics and their use would be beneficial in future comparative studies [91, 89, 90].

4.6 Applicability of Methodology to Problem

The purpose of the methodology was to assess various algorithms for predicting maintenance delays in nuclear power plants. By establishing baseline results and configurations, performing sensitivity analysis by adding noise to the data and reducing the dimensionality, the results were used to develop a framework.

4.6.1 Literature Review

The literature review conducted established the mechanisms in maintenance operations that cause delays and how they relate to maintenance deferral. This part of the study direct coincides with Objectives 1, 2 and 3 as outline in the introduction.

The information gathered here aided in developing the proposed framework.

4.6.2 Data and Baseline

The data selected for this work is similar to ones that can be found in nuclear power plants. The naval propulsion system modeled by the data is in line with data that can be extracted in the desired setting. Both nuclear plants and the CODLAG naval propulsion plant have multiple components that form complex systems. The components of these systems require continual maintenance and many of these components cannot undergo isolated maintenance as they affect the operability of other components in the system. Another similarity in the data is the type of components that are represented. Both the naval plant and nuclear power plants contain diesel generators, turbines and compressors. Both the military and nuclear organizations are strongly supportive of the condition-based maintenance approach. Hence this database, while not directly related, will be able to provide insights into the applicability of machine learning techniques for nuclear applications.

By having a representative data set that closely resembles the type of data found in nuclear power plants, the results of the baseline numerical experiments can be used to gauge the potential performance of these algorithms. These experiments would also serve as a potential starting point for implementing the proposed system in nuclear power plants.

The experiments with the various types of algorithms would also inform which type is suited for this type of data and provide a recommendation for which algorithm would be best for predicting delays.

This part of the methodology directly addresses Objective 4 as outlined in the introduction.

4.6.3 Noise

As mentioned, in nuclear power plants, systems and processes are expected to have a high level of reliability and safety. In the proposed system, the biggest area of concern would be the reliability of both the prediction ability as well as the data received. The results in the baseline experiments comment on the prediction ability given the expected data, however, they do not comment on the sensitivity of the prediction ability with respect to additional noise.

The mechanisms for erroneous data and the reliability of the sensors is out of scope for this work. However, to understand the reliability of the prediction ability under non-ideal conditions, the artificial noise was added to the data. By evaluating the performance based on noisy training data, it will help provide an indication on the type of reliability these algorithms can provide. As mentioned many of the components in power plants are high cost and unique to the application such that replacements are not economically feasible. As such, when a system is used for prognosis and decision making it is important to understand the failure modes and influences that can cause unreliable results.

4.6.4 Dimensionality Reduction:

The PCA reduction in the dimensionality would then provide a sensitivity analysis on how reducing the dimensionality of the training information would affect the error if the prediction models. This in turn would allow for an analysis on accuracy-resource trade off. This method would also give insights on what features are truly relevant in the model and by how much.

4.6.5 Framework Development

Based on the literature review and the experiments performed, two frameworks were developed. These frameworks provide an understanding on how to use delay predictions in maintenance practices. This will aid in understanding where the data comes from, how it is processed and at which stages decisions should be made.

For this work, two different frameworks were proposed to predict maintenance delays depending on the type of data and metrics used. A dual framework approach was used to identify the differences in predicting delays using the representative data and the proposed input data.

5 Results

5.1 Preliminary Data Analysis

5.1.1 Statistical Analysis

The first step in the preliminary data analysis was to have a thorough understanding of the individual features in the data set. The summary of the statistical breakdown of the features can be found in Figure 28.

The key takeaways from the statistical summary showed that both the *Gas Turbine Compressor Inlet Air Pressure* and *Gas Turbine Compressor Inlet Air Temperature* had one constant value throughout the data set and due to this, these features were removed from the data set as it would provide no added benefit.

The statistical summary showed that for all of the input features except for the *lever position* and *ship speed*, did not follow a normal distribution based on their skewness values. With all of them being over a 0.5 skewness threshold to assume normal distribution. To visualize this, box plots and probability density graphs were created. Figure 29 presents the probability density functions of all the attributes and Figure 30 presents the associated box plots. As it can be seen, other than the speed and lever position, none of the input features are centered on the box plots representing the skew. In the probability density graphs, the skewness can also be seen for the input features based on the fitted lines for each of the histogram. All the fits had either multiple peaks or their primary peak was skewed to one end of the distribution indicating that normal distribution assumptions could not be applied.

	Lever Position	Ship Speed	Gas Turbine Shaft Torque	Gas Turbine Rate of Revolution	Gas Generator Rate of Revolution	Propeller Torque	Port Propeller Torque	High Power Turbine Exit Temperature	Gas Turbine Compressor Inlet Air Temperature
count	11934	11934	11934	11934	11934	11934	11934	11934	11934
mean	5.17	15.00	27247.50	2136.29	8200.95	227.34	227.34	735.50	288.00
std	2.63	7.75	22148.61	774.08	1091.32	200.50	200.50	173.68	0.00
min	1.14	3.00	253.55	1307.68	6589.00	5.30	5.30	442.36	288.00
25%	3.14	9.00	8375.88	1386.76	7058.32	60.32	60.32	589.87	288.00
50%	5.14	15.00	21630.66	1924.33	8482.08	175.27	175.27	706.04	288.00
75%	7.15	21.00	39001.43	2678.08	9132.61	332.36	332.36	834.07	288.00
max	9.30	27.00	72784.87	3560.74	9797.10	645.25	645.25	1115.80	288.00
skew	0.0223	0.0000	0.7653	0.5670	-0.1397	0.8070	0.8070	0.5606	0.0000
	Gas Turbine Compressor Outlet Air Temperature	**High Power Turbine Exit Pressure**	**Gas Turbine Compressor Inlet Air Pressure**	**Gas Turbine Compress or Outlet Air Pressure**	**Gas Turbine Exit Gas Pressure**	**Turbine Injection Control**	**Fuel Flow**	**Gas Turbine Compressor Decay**	**Gas Turbine Turbine Decay**
count	11934	11934	11934	11934	11934	11934	11934	11934	11934
mean	646.22	2.35	1.00	12.30	1.03	33.64	0.66	0.98	0.99
std	72.68	1.08	0.00	5.34	0.01	25.84	0.51	0.01	0.01
min	540.44	1.09	1.00	5.83	1.02	0.00	0.07	0.95	0.98
25%	578.09	1.39	1.00	7.45	1.02	13.68	0.25	0.96	0.98
50%	637.14	2.08	1.00	11.09	1.03	25.28	0.50	0.98	0.99
75%	693.92	2.98	1.00	15.66	1.04	44.55	0.88	0.99	0.99
max	789.09	4.56	1.00	23.14	1.05	92.56	1.83	1.00	1.00
skew	0.4257	0.7069	0.0000	0.6285	0.7730	0.8968	0.9995	0.0000	0.0000

Figure 28: Summary of statistical information of Turbine Decay Prediction Data Set.

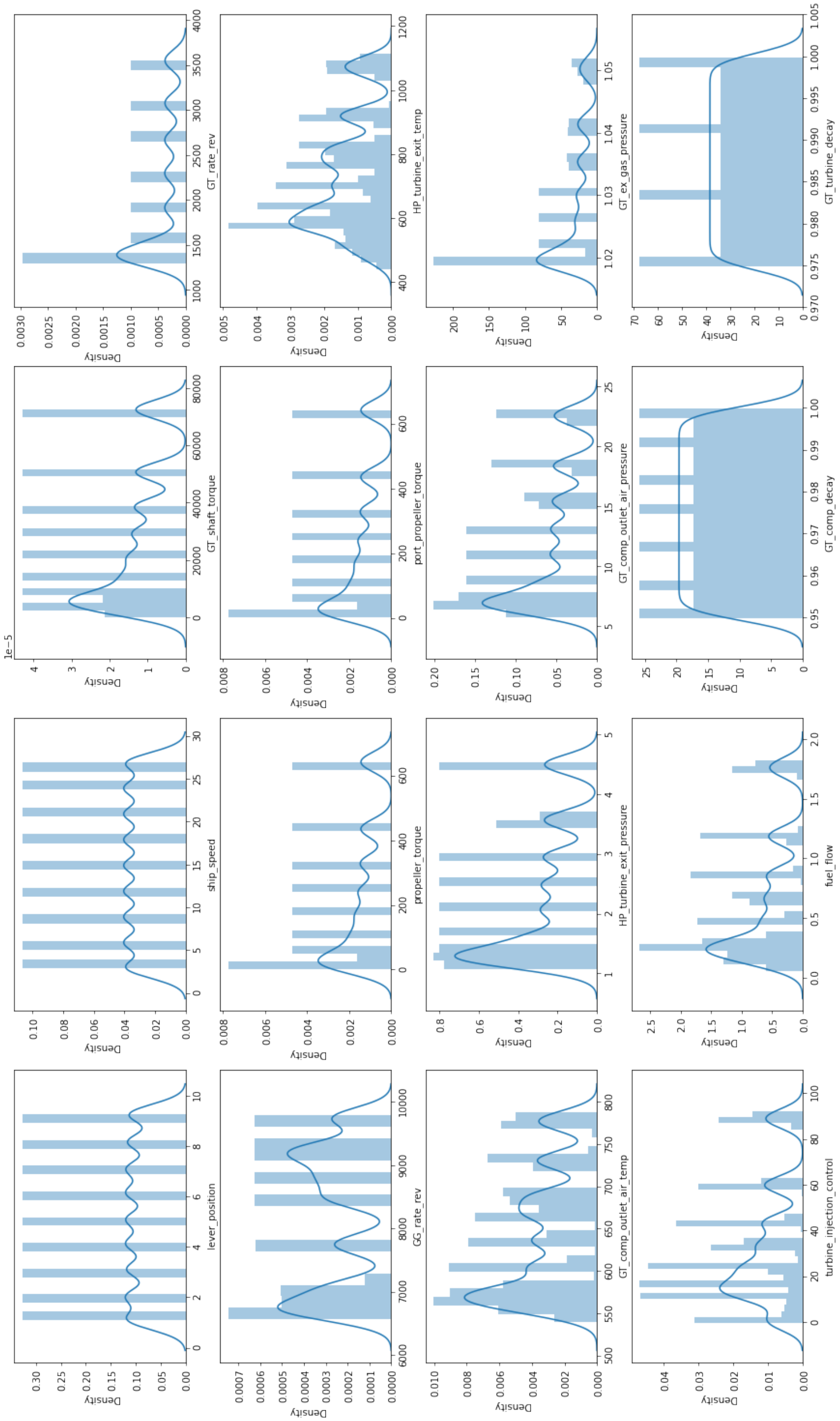

Figure 29: The probability density function of all the attributes in the data set.

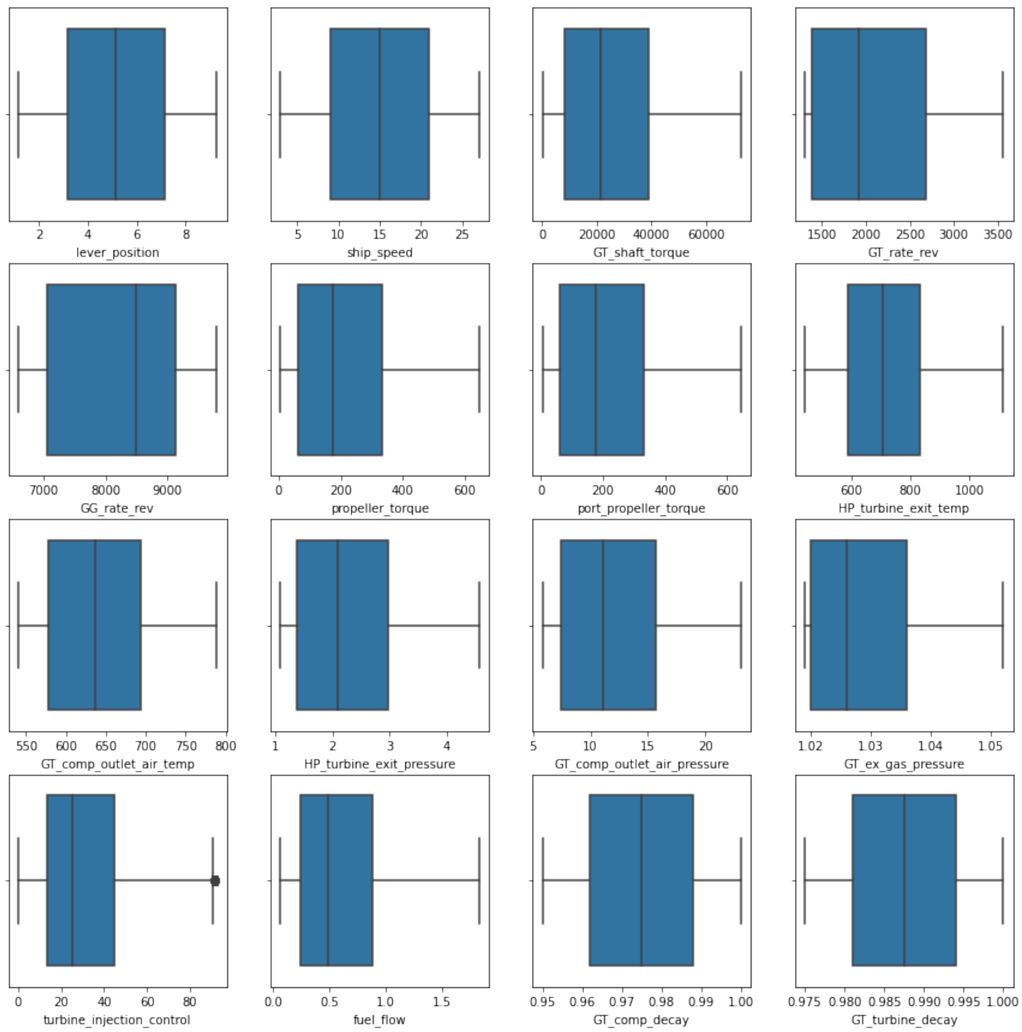

Figure 30: Box Plots for all the attributes in the data set.

This information was important because it indicated that some numerical standardization or feature scaling was required in order for various algorithms to work. Many algorithms work best when all the features are centered around a common mean which this data was not. Also, to ensure that the scaling of different features do not affect disproportionately influence the model, feature scaling was performed. As mentioned in the methodology, *SKlearn* Standard Scalar was used to standardize the data.

5.1.2 data set Labels

To better understand the prediction values (Compressor and Turbine decay), the labels were first plotted as seen in Figure 31. This showed that the for each of the sampled compressor decay values, different turbine decay values were recorded. It is to be noted that the data set samples from [1:0.95] and [1:0.975] for the compressor and turbine decay respectively so values from brand new install (decay of 0) are not reported, only end of operation cycle.

It is to be noted that the turbine decayed at a quicker rate than the compressor. There were 52 turbine decay cycles or the one compressor decay cycle. Also, for every turbine decay cycle, the compressor would decay by one increment meaning that the maintenance of the turbine results in a slight degradation of the compressor.

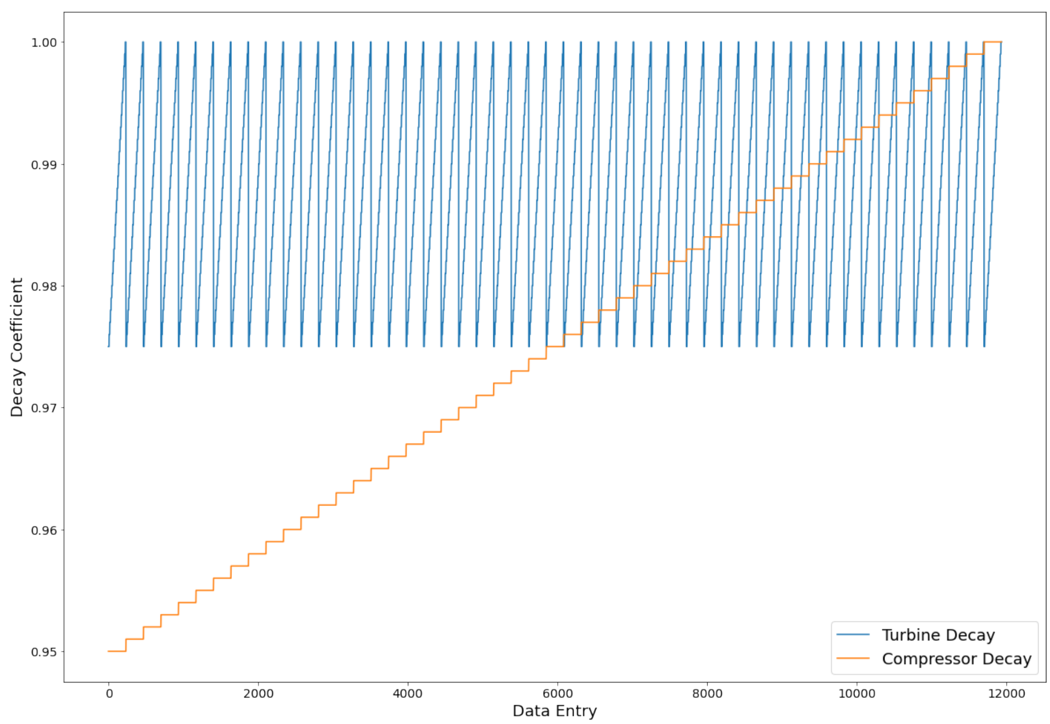

Figure 31: Plot of all the compressor and turbine decay labels in the data set.

As seen in Figure 32, the number of samples for the labels is approximately uniformly distributed. This means that the ratio of all values in both the training and testing splits would be equal.

In this work, a train-test split ratio of 80% to 20% was selected. Though the data set has a significant selection of data, it is still relatively small and would benefit from more training data. Also, the probability density graphs of the labels showed that the value distribution is relatively uniform and the training and testing set would have the same proportion of prediction values.

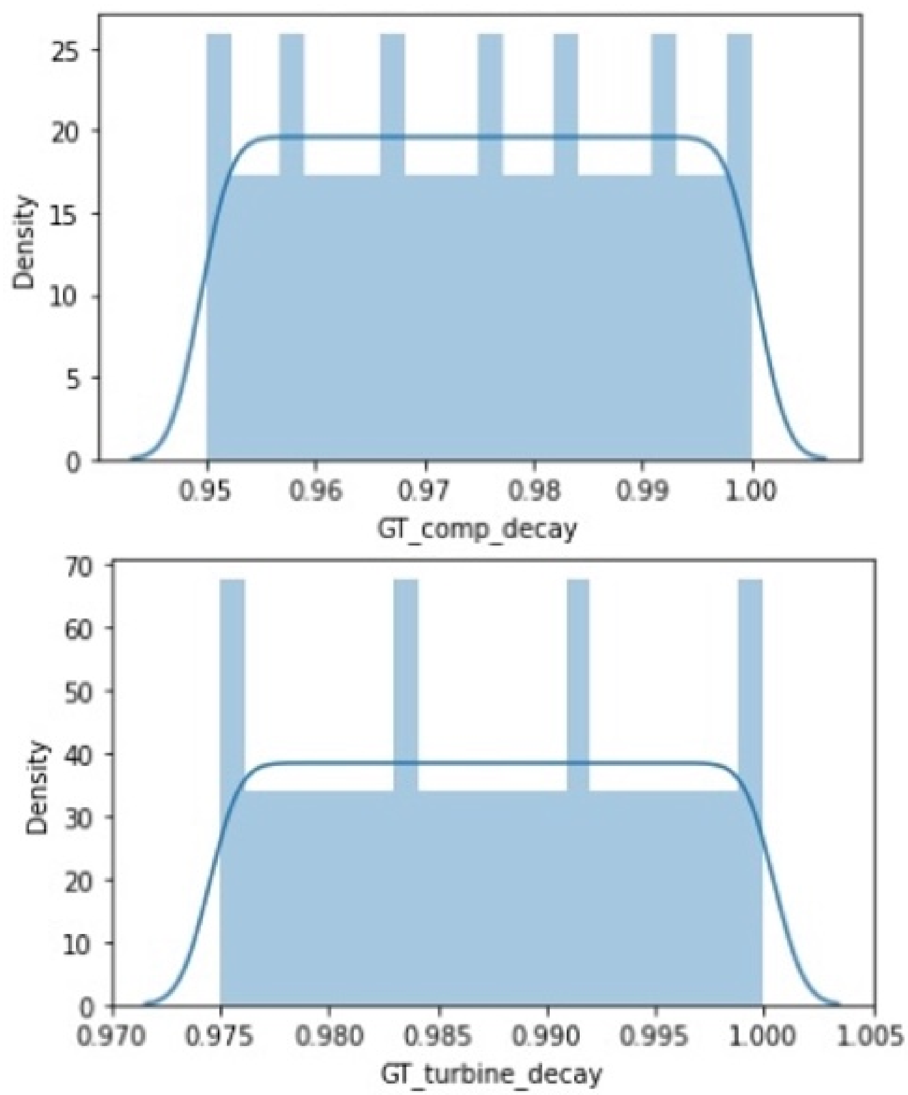

Figure 32: Probability density graph of the compressor and turbine decay coefficients.

5.1.3 Variable Correlation

As seen in Figure 33, the *Pearson Correlation Coefficients* were determined for all of the features and labels against each other in the data set. It is to be noted that in the figure, the diagonal plotted the correlation for each feature to itself, thus is a perfect correlation of 1. The correlations are mirrored across the diagonal of the table.

As it can be seen, all of the features are highly correlated to each other. However, there is no direct relation between the input features and the labels. This is expected as the decay value is a ratio of other engineering units (the flow rate through the engine). This lack of explicit relation provided the opportunity to implement various algorithms to be able to map the input variables to the prediction labels.

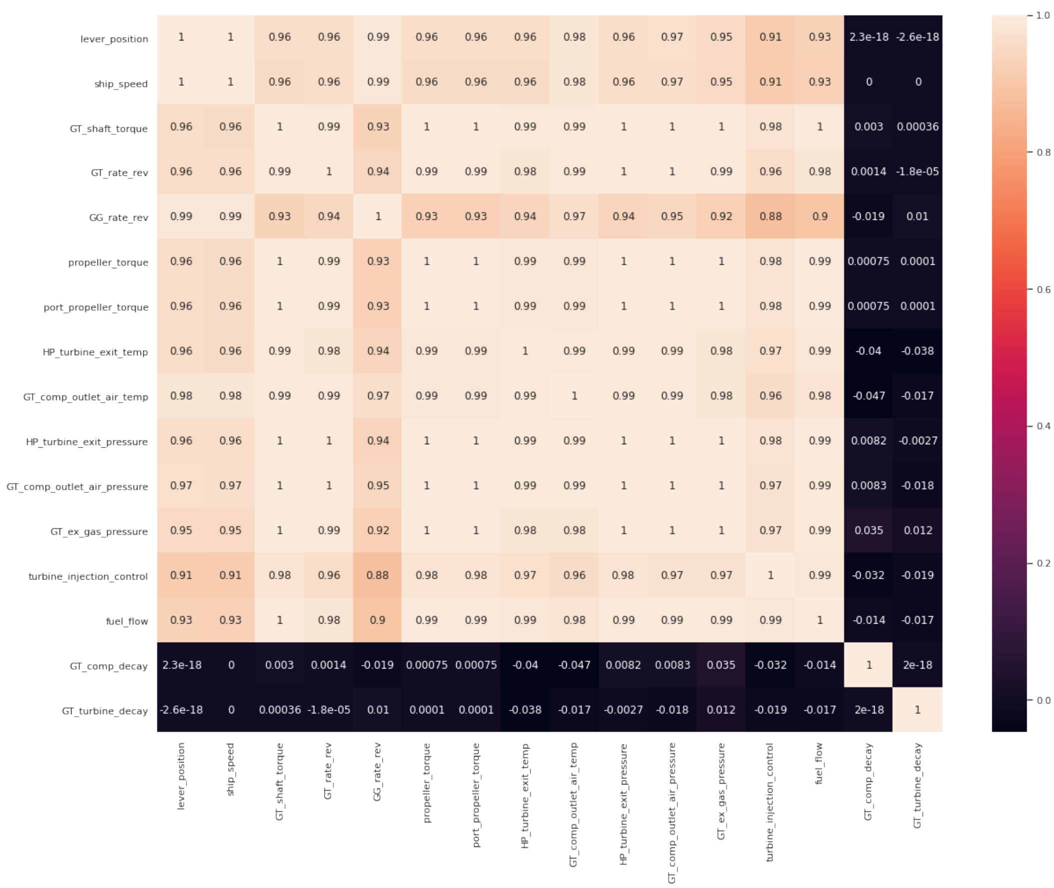

Figure 33: Correlation table for all of the features in the data set using Person Correlation Coefficient.

5.2 Prediction Results

One important aspect of this work was to comment on the prediction ability of the various methods and determine what configurations would be useful for predicting delays.

Some of the primary considerations included the types of algorithms, the inclusion of noise into the data, and prediction ability when removing data. The following are the results for the various algorithms and data manipulation tested.

The primary metric used to evaluate the results was the Mean-Absolute-Error (MAE). The metric was used as it provides an average error range in which the prediction is correct. This range serves as a confidence interval which will be discussed in the framework section of the results. This metric was also selected because it equally weighs all the errors in the calculations.

The Mean-Squared-Error squares all the error values and this causes the high error values to have stronger influence on the average error and the lower values to have a smaller influence. Due to this behaviour in the metric, it was also used because it provided an indication on how tight or spread apart the errors are from the label value. A high MSE meant that there is more of a spread whereas a lower MSE indicated prediction values were much closer to the label values.

For the *SKlean* algorithms, the R^2 value was also used to determine the quality of the fit.

5.2.1 Algorithms

This section outlines the best configurations for the various algorithms selected. These configurations were used as a baseline for results without noise and further data manipulation. These configurations were then used with noisy data and PCA dimensionality reduction.

SKlearn: The following are the best configurations found for the SKlearn algorithms used based on data with no noise inclusion:

- **Support Vector Regression:** Regularization parameter (C) over 1, Radial Basis Function as the kernel function and a epsilon value of 0.001. Standardization of the input data is also required for an accurate prediction. It is to be noted that Regularization parameter (C) of 0.5, 0.75, 1, 2 and 4 were tested but values under 1 either resulted in low MAE or no fitting at all. Values over 1 resulted in diminishing returns.

- **Decision Tree:** The only parameter tested for Decision Tree was the minimum number of samples for a decision split. It was found for this data that splits of 2,5 and 10 all had similar magnitudes in their MAE values, however, splits of 2 and 5 had the lowest degree in their error and lowest MSE. Due to the similar results, 5 splits are recommended as it reduced the tree size and less decision splits are made, serving to be more useful in over-fitting. It was also noted that standardizing the data did not affect the magnitude of the error but did increase the degree in the error. To validate the results, the R^2 value was determined, and the standardized runs had a score of approximately 0.1 whereas the non-standardized runs had a score of up to 0.98. This informed that the fit of the standardized input data does not accurately correlate to the prediction value, thus, it is recommended not to standardize for this algorithm.

- **K-Nearest Neighbour:** the number of neighbours was tested between 2,5 and 10 neighbours. Both 2 and 5 neighbours harvested the lowest MAE score, however, for this work 5 neighbours were used. For standardization, the results showed that the feature scaled data had twice the MAE error, however, the magnitude of the error is 10^{-3}. The standardized data had much tighter error bounds with the MSE being a magnitude lower (10^{-6}). The non-standardized data had a spread in the error of up to 5%. This means a label of 0.95 in the compressor decay coefficient was being reported as 1.00. Due to spread in the error value, standardization is recommended for this algorithm.

- **Multiple Regression:** As this algorithm uses Ordinary Least Squares, no parameters were altered for this algorithm. Standardization did not provide any considerable difference in prediction capabilities, however it was used for a better comparison to the multiple-regression network in the *Tensorflow* models.

Tensorflow Networks: The following are the best configurations for the Tensorflow networks based on data with no noise inclusion. It is to be noted that standardization was required for all the networks to get consistent reproducible results.

- **Regression Network:** For the Tensorflow regression network, various learning rates and training cycles were used. It was found that a learning rate of 0.001 was the most stable during training and after 300 epochs there were diminishing returns on the MAE, thus, making these the recommended configurations.

- **Deep Neural Networks:** The three different proposed network designs were tested using different learning rates and training cycles. Like the regression network, the most stable and reproducible results were found using a learning rate of 0.001 and after 300 epochs, there were diminishing returns on the MAE. However, for the network design, the network with 2 intermediate dense layers of 14 and 7 nodes respectively and the network with 3 intermediate dense layers of 21, 14 and 7 nodes respectively performed similarly. The network with the 3 intermediate dense layers was selected it had the smallest MAE every time and there was difference in computational time.

For all the networks, the training was also performed at 1000 and 2000 epochs but would always converge at around 300 epochs with diminishing returns. For all networks in all tests had low computational requirements at around 3 epochs per second. Figures 34 and 35 present a sample of the training behaviour of both the regression network and the deep neural network; this with respect to the turbine decay. Similar behaviours were found with the compressor decay. Both the MSE during training and MAE were plotted. It is to be noted that the MSE is a squared function thus it is in higher units, thus the difference in the curves. The testing MAE and MSE were also plotted and it can be seen that they correlate to the training values.

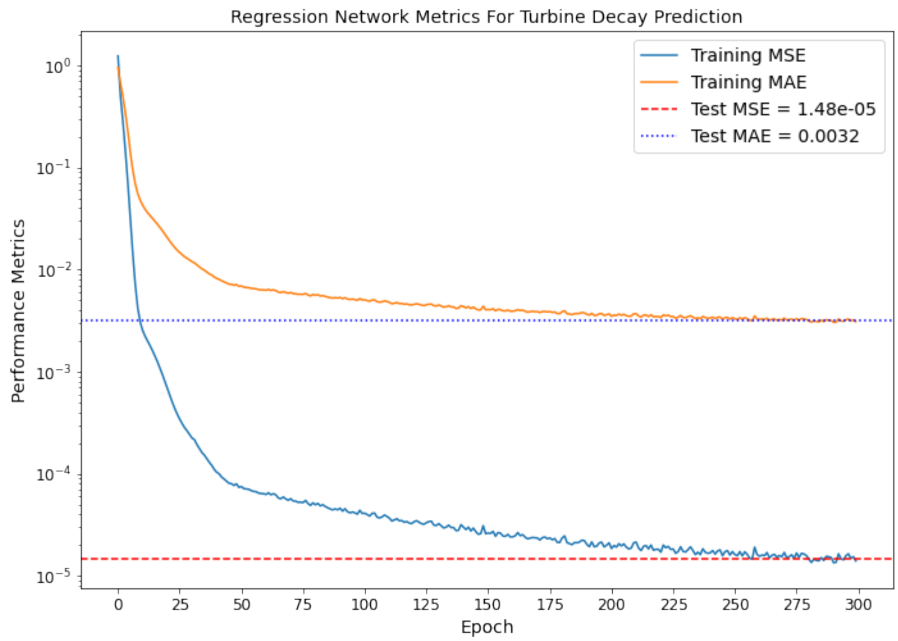

Figure 34: Training results of regression network over 300 epoch and the associated testing results.

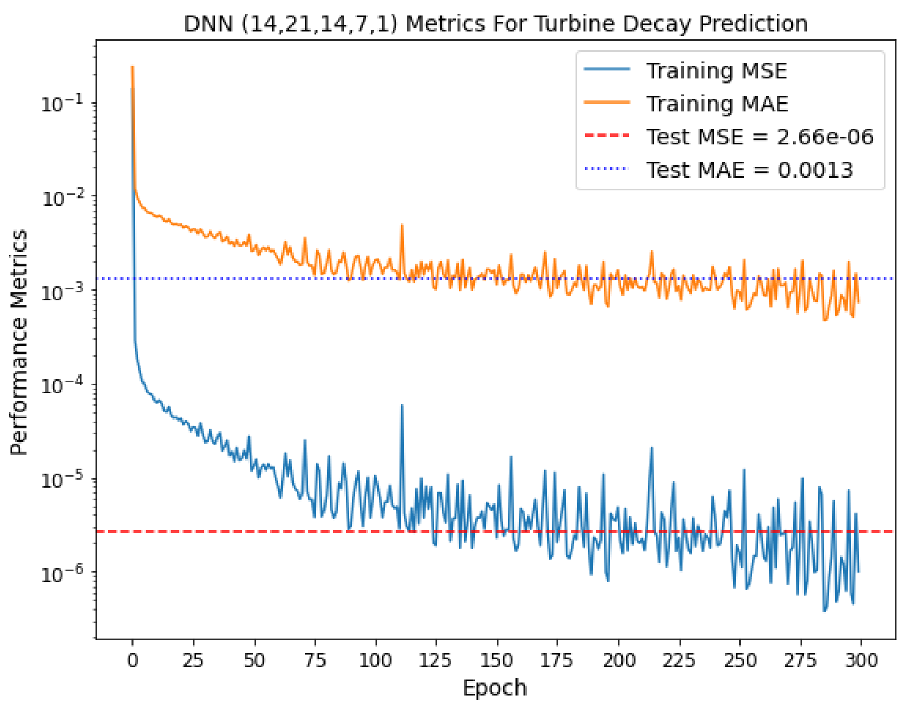

Figure 35: Training results of the deep neural network over 300 epoch and the associated testing results.

For all the runs of noisy and non-noisy data, the regression network always converged in a smooth fashion whereas the DNNs would converge but oscillate to a higher degree around a central point. For example, in Figure 35, the test MSE was approximately the central point in the training, however after 200 epochs the range of the MSE went from 10^{-6} to 10^{-5}. This behaviour past 300 epochs continued and depending on where the training ends, varied the magnitude of the test results.

5.2.2 Results: Baseline

Table 3 presents the best results for each of the algorithms tested. In blue are the best results for the given trial.

As seen in the table, for the compressor decay, the best results were achieved using the Support Vector Regression, Decision Tree and the Deep Neural Network. It is to be noted that the performance differences between the three algorithms are negligible.

The compressor prediction was between values of 0.95 and 1.00, reported to 3 decimal places. This means that most of the baseline models are accurate within the lowest reported unit. After running and plotting all of the predictions, it was established that a model with a good prediction is one where the MAE is lower than $1 * 10^{-2}$ and the MSE is lower than $2.5 * 10^{-6}$. This will be discussed in the next chapter.

For turbine decay prediction, the results were similar to that of the compressor decay prediction. One difference found was that the Ordinary Least Squares regression algorithm performed just as well as the other algorithms. For the decision tree, in the turbine decay prediction, the MAE was an entire magnitude lower than that of the other algorithms.

Figures 36 (compressor) and 39 (turbine) plotted the test labels and the prediction labels for all the *SKlearn* algorithms. It is to be noted that for visual fidelity, after the calculations, the plots ordered the label-prediction pairs in increasing order with respect to the true labels. Figure 37 (compressor) and 40 (turbine) plotted the residuals between the prediction and the label as well an average residual in red. Figure 38 (compressor) and 41 (turbine) plotted the predictions and residuals for both the regression network and the best deep neural network.

As seen in the residuals for both the compressor and turbine, the SVR and the DNN had the tightest spread in the residuals. This is a desirable outcome as the models are less susceptible to outlier errors. Though the mae of the decision tree for the turbine decay was the lowest out of all the runs, it can be seen that there are outliers that extended up to ±2.5%. Though the number of outliers is far and few between when compared to the 2000+ test points, it is something to consider when more features and data points are added.

Another interesting behaviour observed with the data was with the high values, close to a full decay of 1, with the KNN runs. Especially in the turbine decay where there are 52 decay cycles, it is noted that the predictions closer to 1 were all under-predicted. The proposed reason for this is that, while the component is still on a trajectory of decay, there is a constant rise in the decay graph as seen in Figure 31. However, as the decay cycle reaches its end, the 5 nearest neighbours involved four values closest to 1 and then a value of 0.975 which is the decay cycle reset, as a result there is a significant drop in the average, causing the lower prediction. This factor is important because it indicates that KNN algorithm has a particular behaviour in end of cycle predictions which under reports the decay value. This is particularly of concern if the model was to be extrapolated to prediction ranges from 0 to 1. In worst case scenario, the decay would be reported at 0.8 which misrepresented the true behaviour.

Another observation with the runs for the turbine decay was the similarities in the OLS regression and the regression network. Both fared essentially the same results and the same residuals. Even for the compressor decay, the two models were very similar to each other. This is useful because OLS regressions are only useful for inputs with linear combinations whereas neural networks are adaptable to various combinations. Whereas OLS regression is fixed in structure, the neural network is useful as the network design can be adjusted to increase prediction capabilities. In this application, using the regression network is useful because if the prediction performance is not what is expected, it is an indication in the OLS performance as well. The very same model can be adjusted to a deep neural network to then increase the performance. This can be seen in Table 3 and the associated figures where the DNN performance was always better.

Table 3: Best Results of Algorithms with No Noisy Data: Baseline Results

Compressor Decay	Mean-Absolute Error	Mean-Square Error
Support Vector Regression	0.0011	1.60E-6
Decision tree	0.0010	3.30E-6
K-Nearest-Neighbour	0.0021	7.38E-6
Multiple Regression	0.0046	3.42E-5
Multiple Regression Network	0.0062	5.69E-5
Deep Neural Network	0.0014	2.83E-6
Turbine Decay	Mean-Absolute Error	Mean-Square Error
Support Vector Regression	0.0013	2.52E-6
Decision tree	0.0006	1.62E-6
K-Nearest-Neighbour	0.0026	1.05E-5
Multiple Regression	0.0017	5.13E-6
Multiple Regression Network	0.0019	5.92E-6
Deep Neural Network	0.0013	2.66E-6

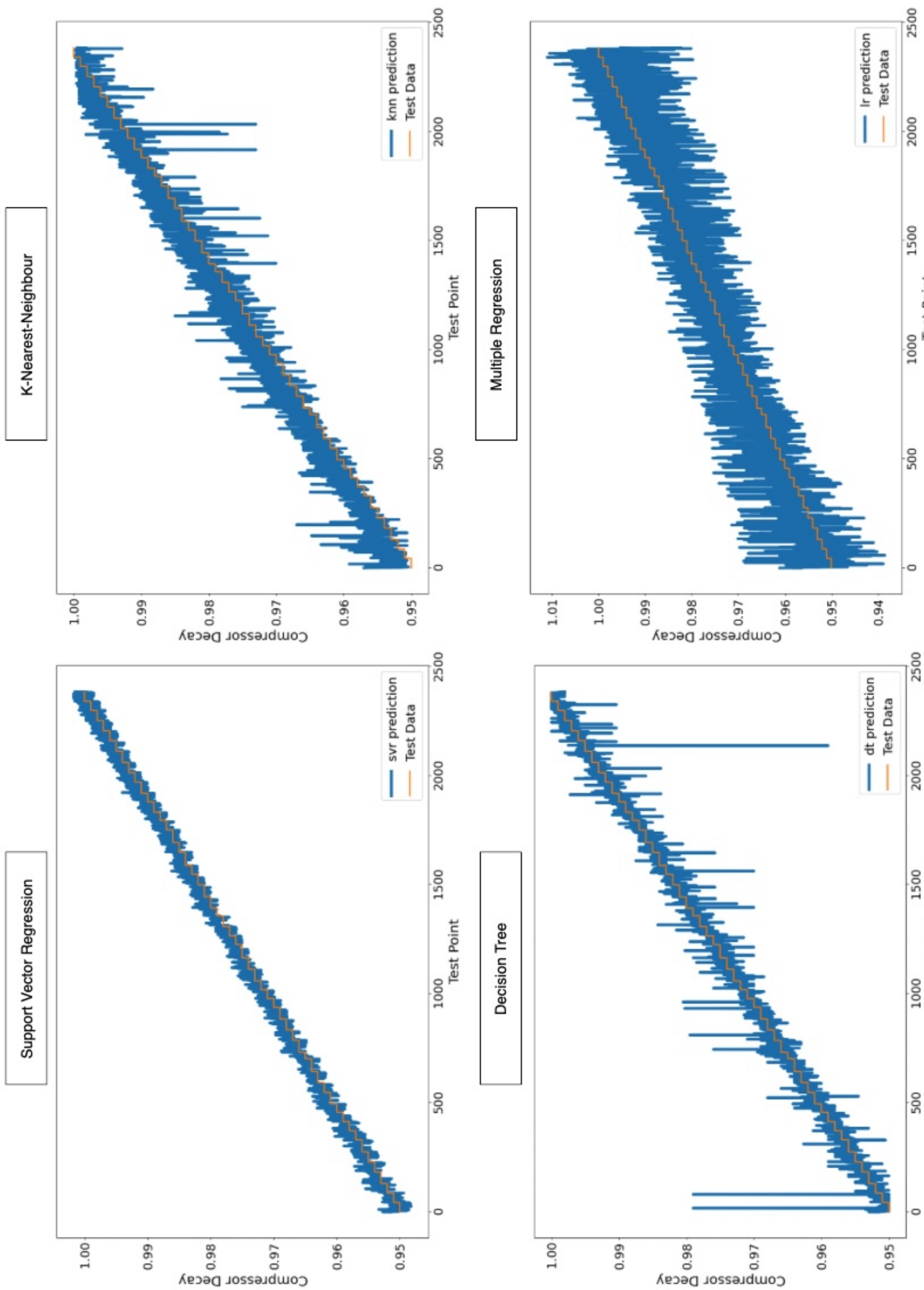

Figure 36: Baseline compressor decay prediction results *SKlearn* algorithms.

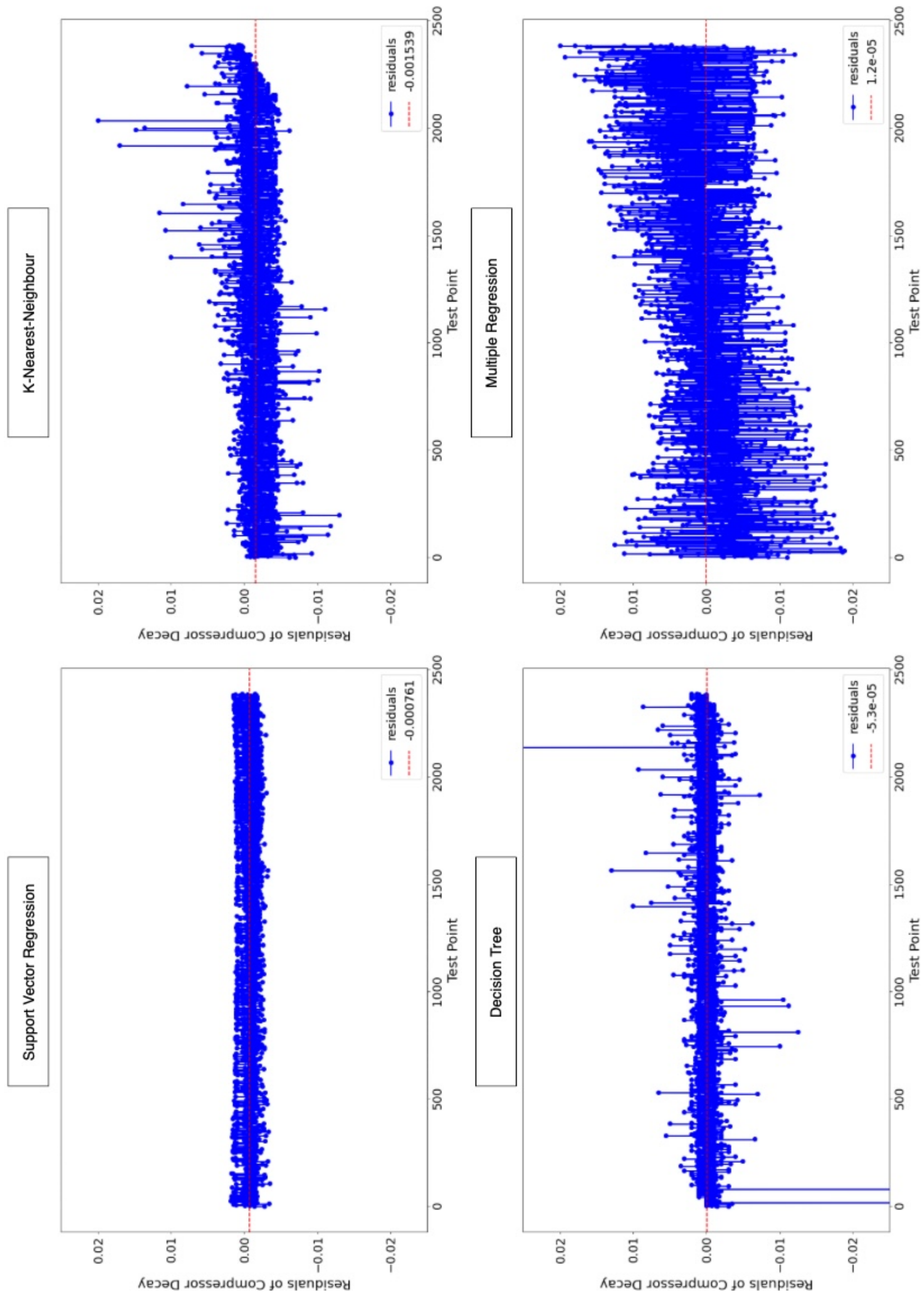

Figure 37: Baseline compressor decay prediction residuals with the red line being the average residual for *SKlearn* algorithms.

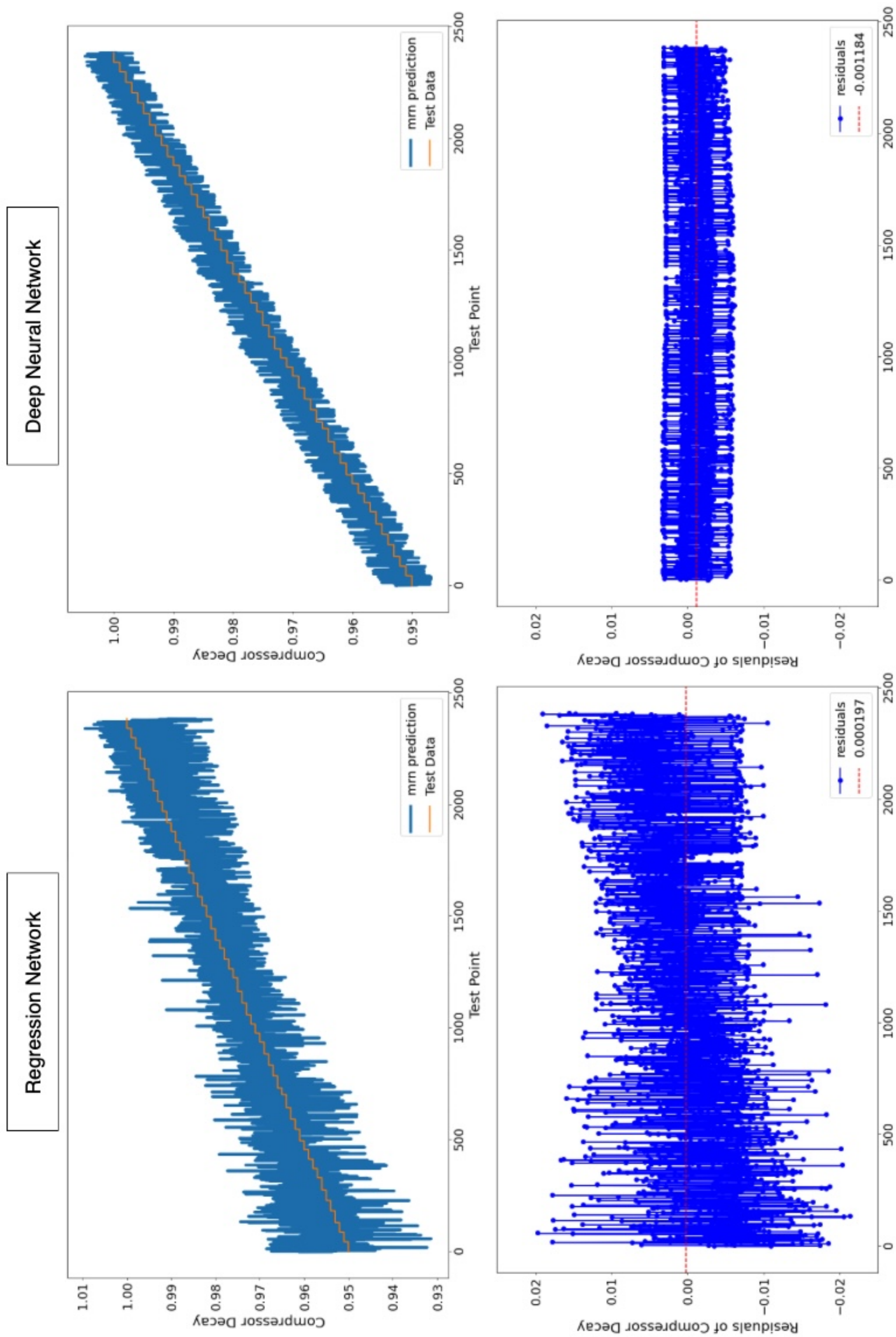

Figure 38: Baseline compressor decay prediction results for *Tensorflow* network models.

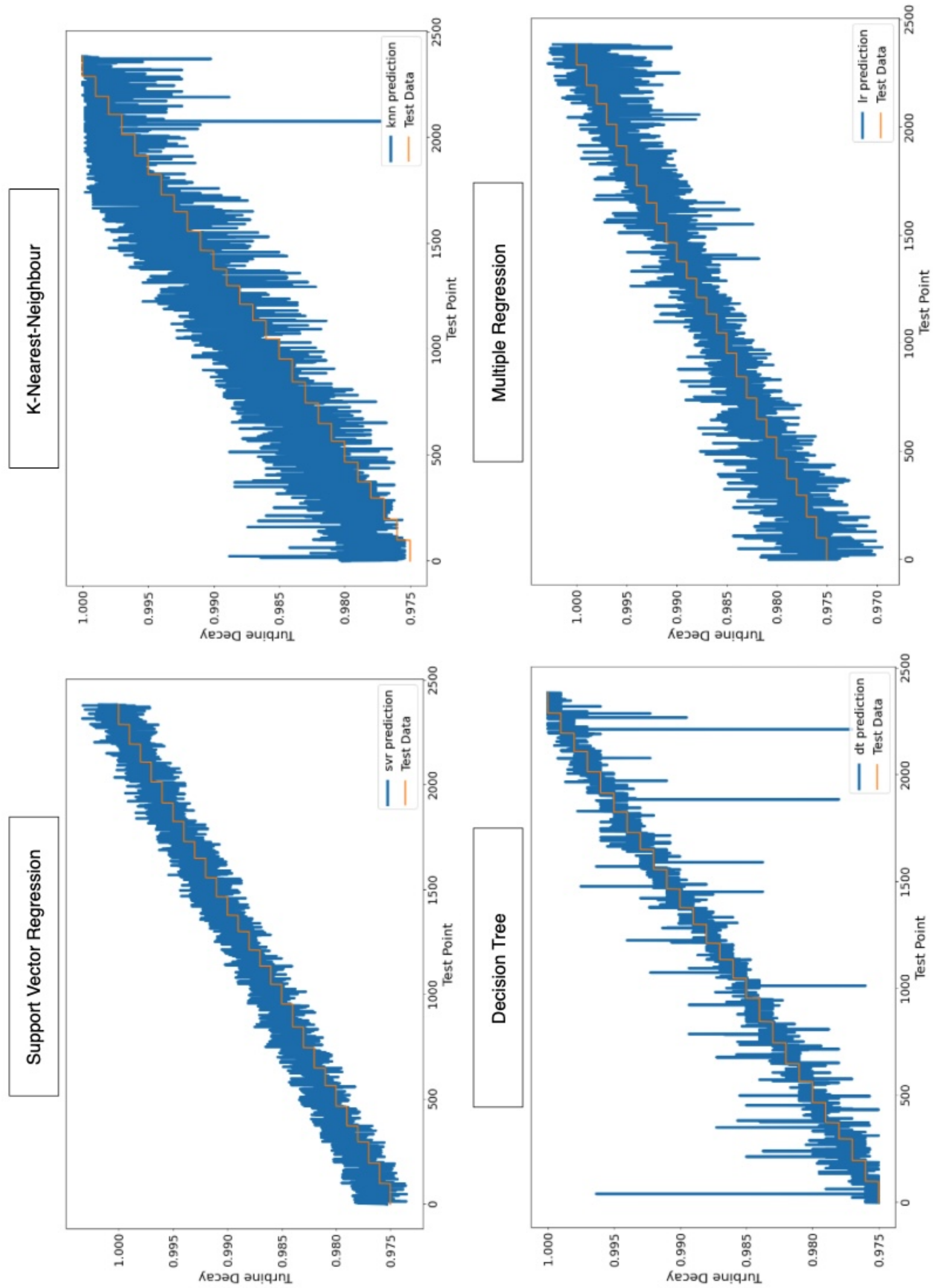

Figure 39: Baseline turbine decay prediction results *SKlearn* algorithms.

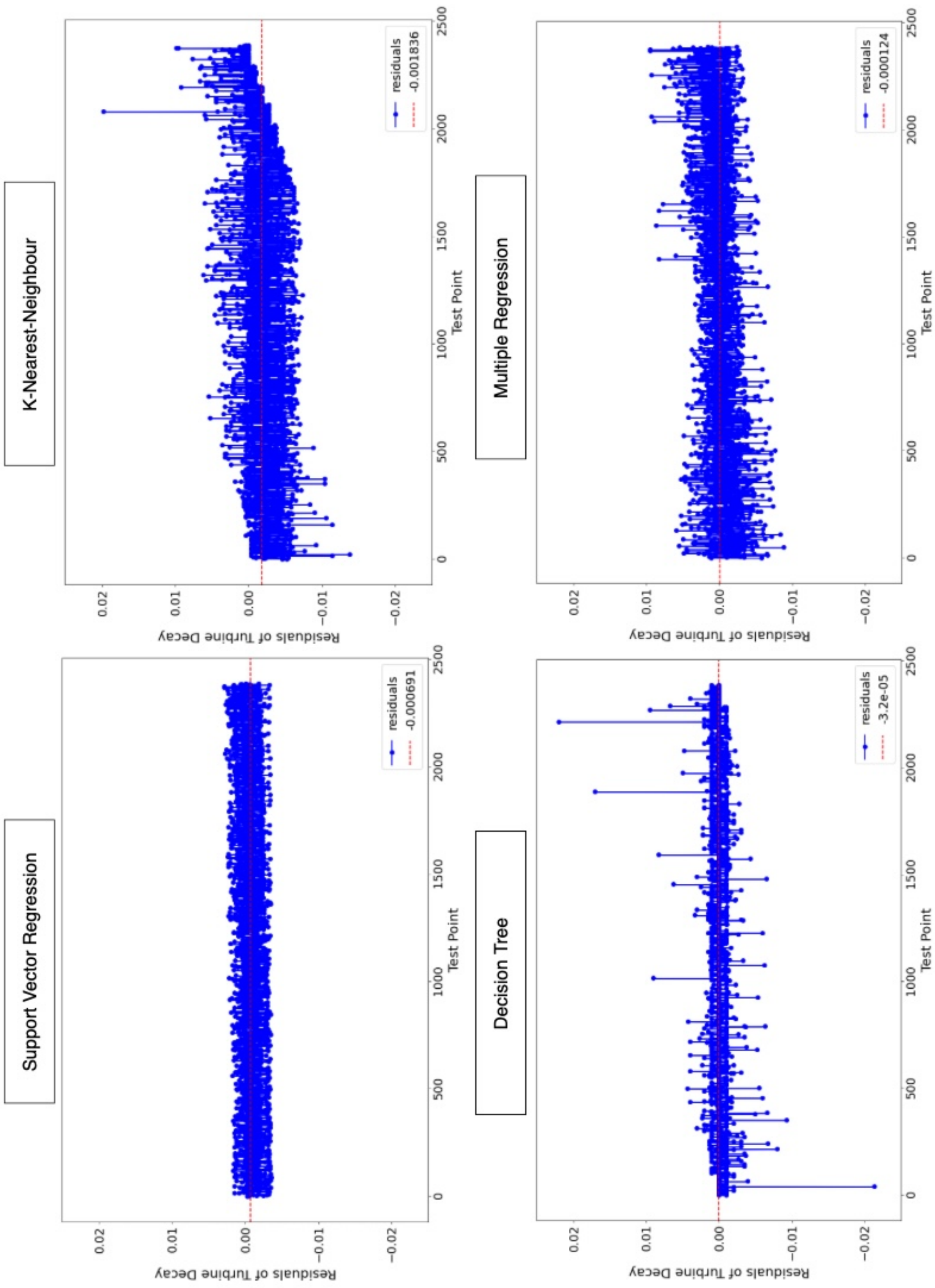

Figure 40: Baseline turbine decay prediction residuals with the red line being the average residual for *SKlearn* algorithms.

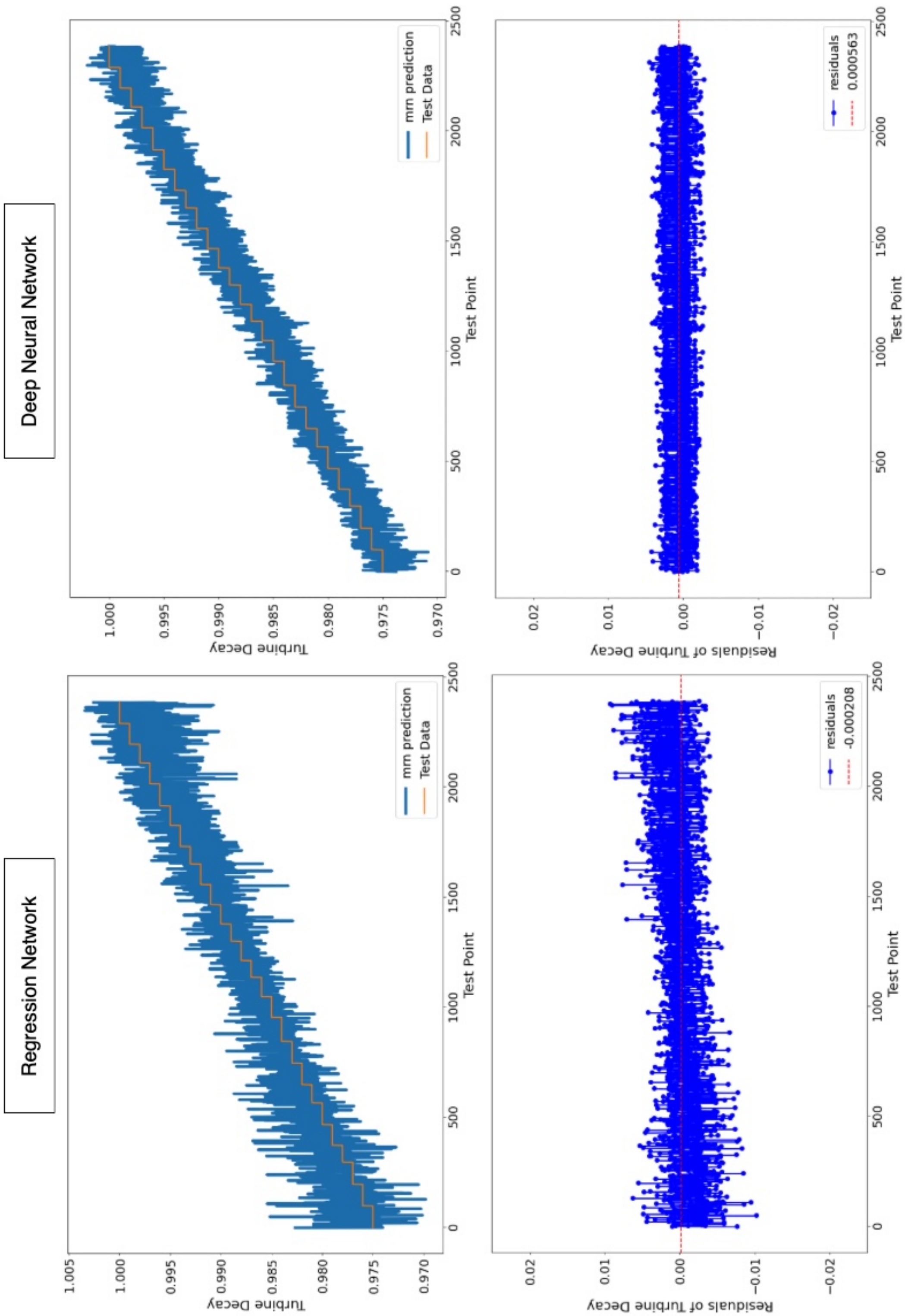

Figure 41: Baseline turbine decay prediction results for *Tensorflow* network models.

5.2.3 Uniform Noise on Test Set Results

This set of experiments added noise of various ranges to only the test set of the algorithm. This was completed in order to determine the sensitivity of trained models to noise data. As seen in the results in Table 6 and Figures 42 and 43, small amounts of noise do not affect the prediction ability. However, noise as little as $\pm 1\%$ makes the prediction ability of the algorithms not suitable for both the turbine and compressor decay predicitons.

It is to be noted that the Deep Neural Network performs the worst at higher noise ranges.

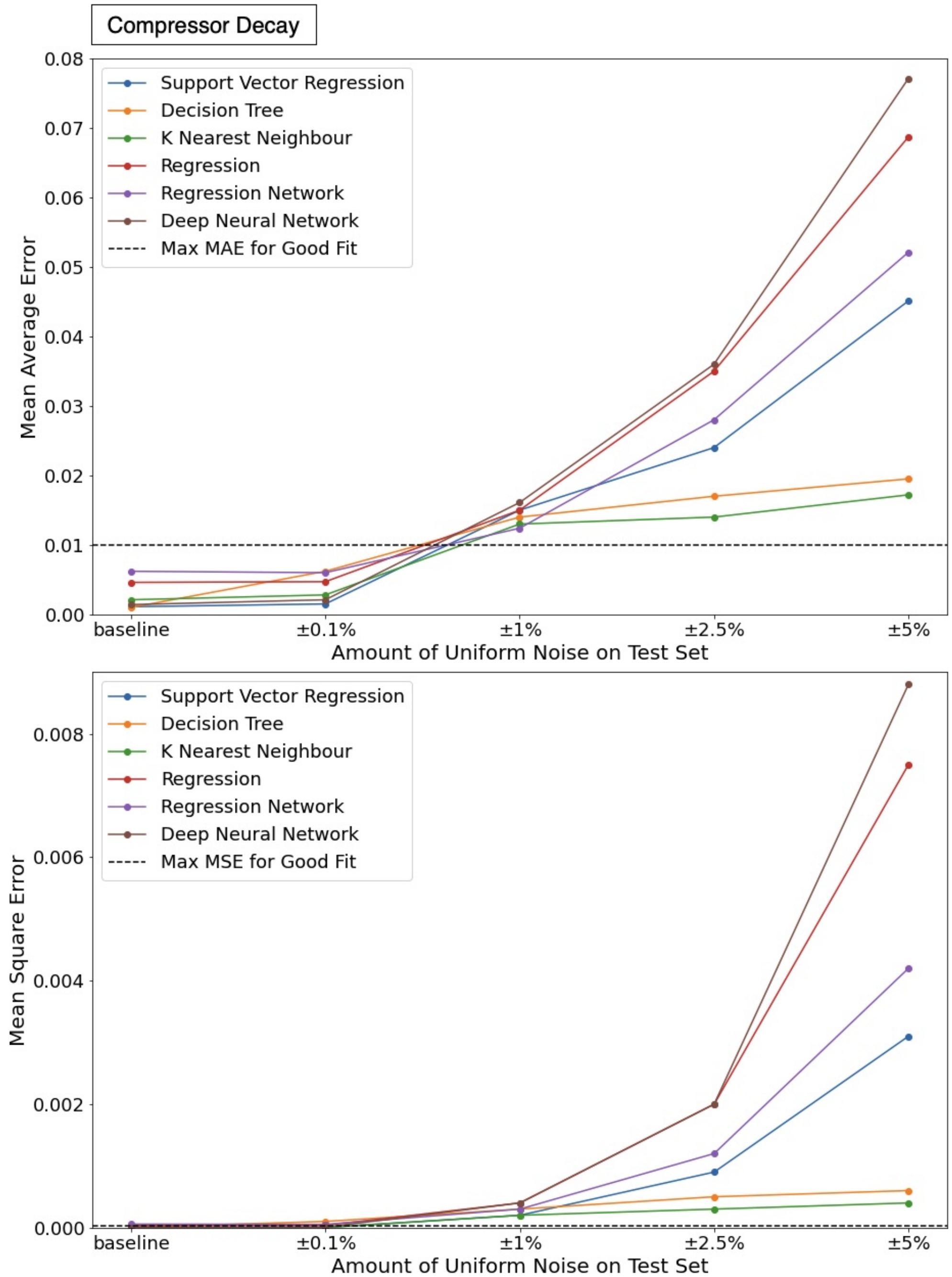

Figure 42: Compressor: mean average error and mean square error of the algorithms with various amounts of uniform error.

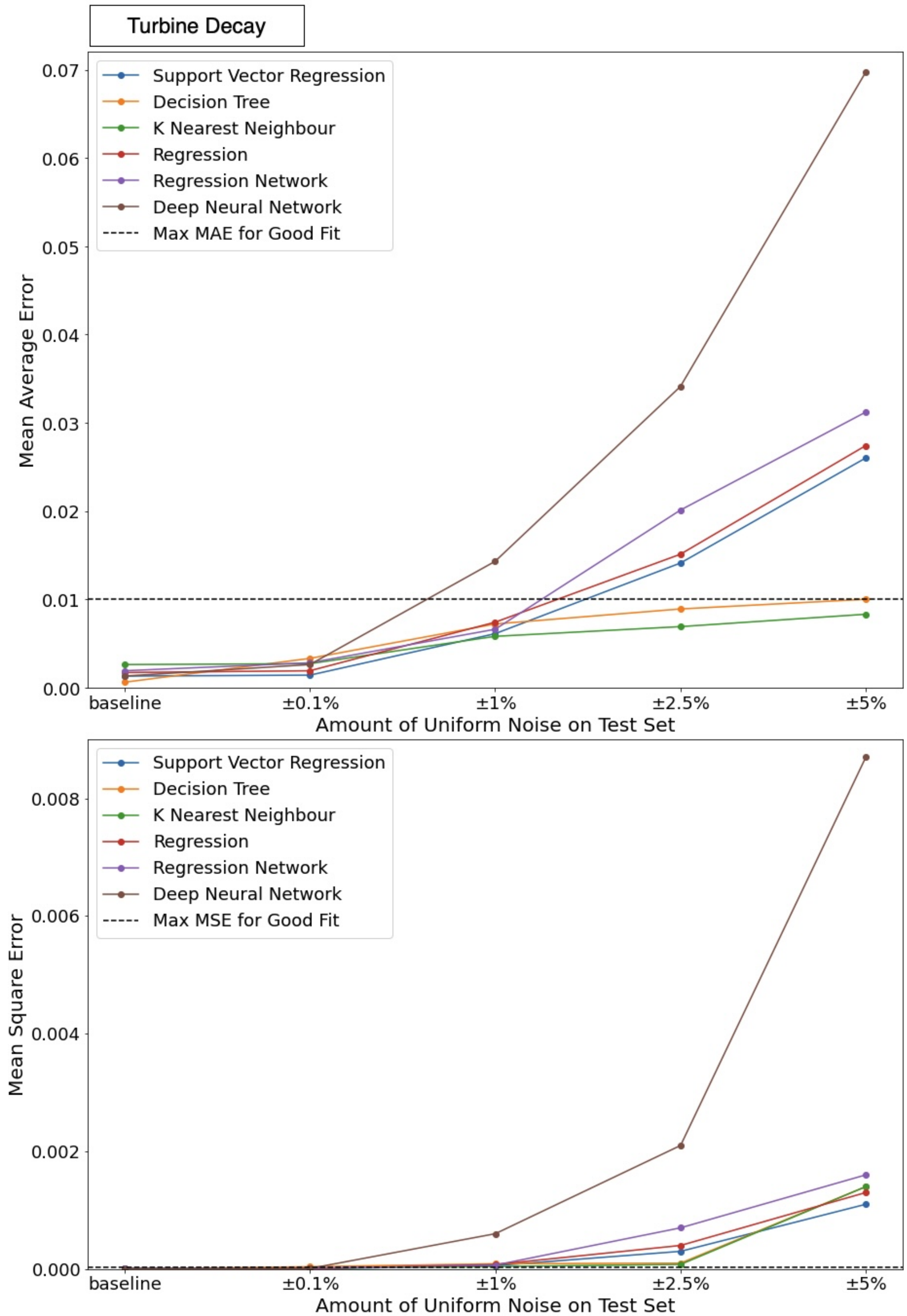

Figure 43: Turbine: mean average error and mean square error of the algorithms with various amounts of uniform error.

5.2.4 Uniform Noise Results

The next set of experiments performed were running the baseline configurations of the algorithms with noisy data on both the test and training sets. This is a common method to improve prediction capabilities with respect to noisy data.

The first set up was applying a range of noise to the entire database. As seen in Table 7, multiple runs with noise of $\pm 0.1\%$, $\pm 1\%$, $\pm 2\%$ and $\pm 5\%$ were uniformly added. A random seed of 14 was used. It is to be noted that the noise range was that of the input data to their respective scale. Any feature scaling was performed after noise was added. The labels were left as is.

Uniform data was used to simulate a scenario where a hypothetical error in the data transmission would affect the entirety of the data set for training.

As seen in Table 7 and Figures 44 and 45, the $\pm 0.1\%$ runs had no affect and at $\pm 1\%$, there is a slight increase in the error however the error is still in the range of the lowest reported significant figure. Beyond that, the error was significantly higher and the prediction was not accurate. It is to be noted that even though for most inclusions of noise, the MAE is lower than the designated threshold, when analyzing the MSE it was noted that the fits were not good. This can be seen in both the compressor and turbine decay predictions. Though it is noted that the turbine decay predictions, most algorithms were within tolerance at 1% noise.

For certain algorithms with an additional $\pm 0.1\%$ error, the performance improved, such as the KNN model for the turbine decay prediction. This may be an anomaly as the error may have biased the input data to favour the algorithm however the general trend was that more noise decreases the performance of the models.

Upon further exploration of the results, it was noted that for both the uniform and selective noise runs, the MSE was important as the MAE alone could be a false positive indicator of performance. For example, Figure 46, presents the prediction results of the Decision Tree model with $\pm 5\%$ error on the data and as noted, the MAE is in a range that is considered good. However, the MSE is a magnitude lower than that of the baseline results. When the prediction is plotted, it is noted that the prediction was arbitrary and was not a good fit. The lowest values were reported as the highest and vice versa. This result indicated that any MSE value higher than $2 * 10^-5$ is on average a bad fit and is similar to behaviour as seen in Figure 46.

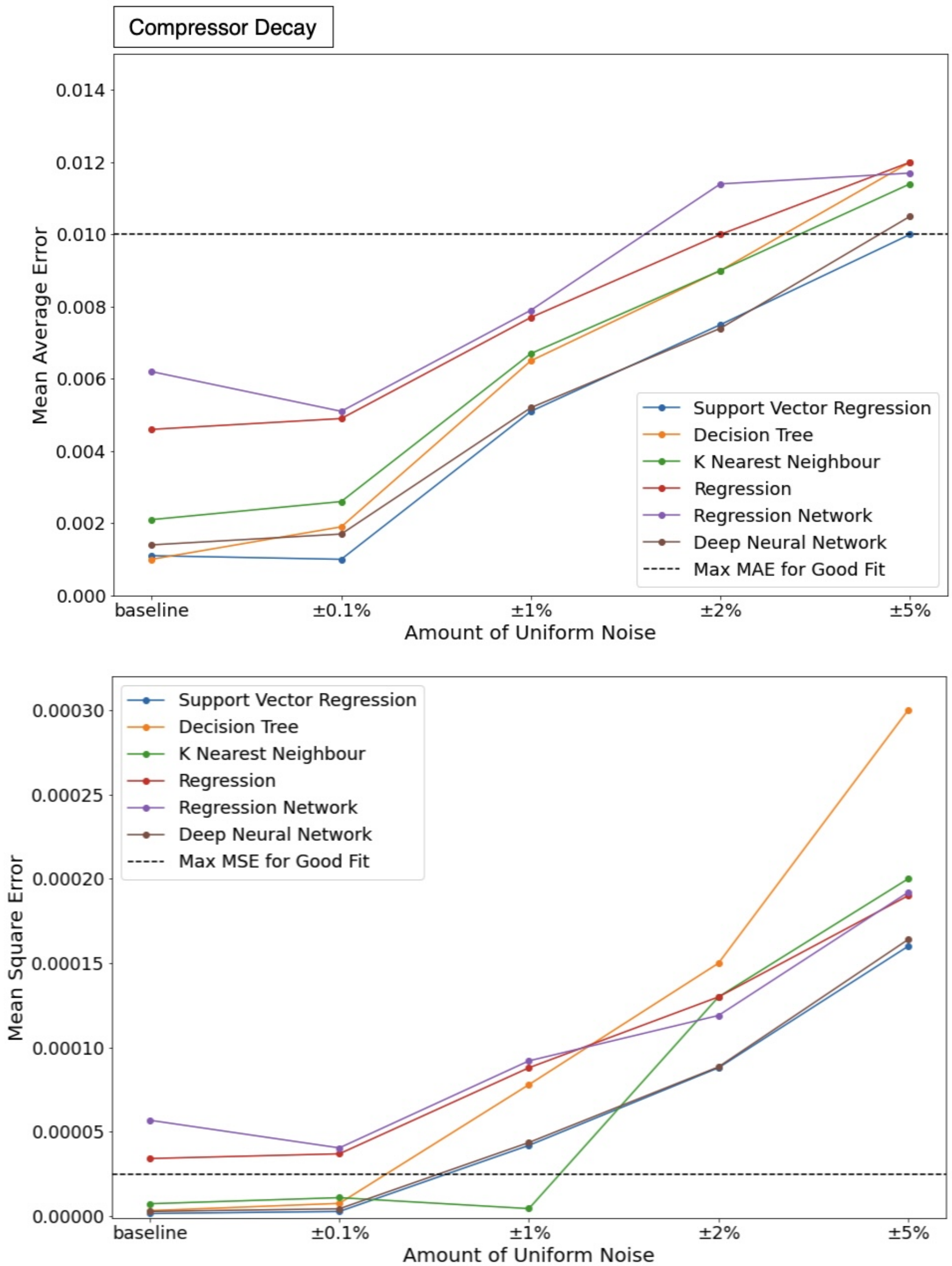

Figure 44: Compressor: mean average error and mean square error of the algorithms with various amounts of uniform error.

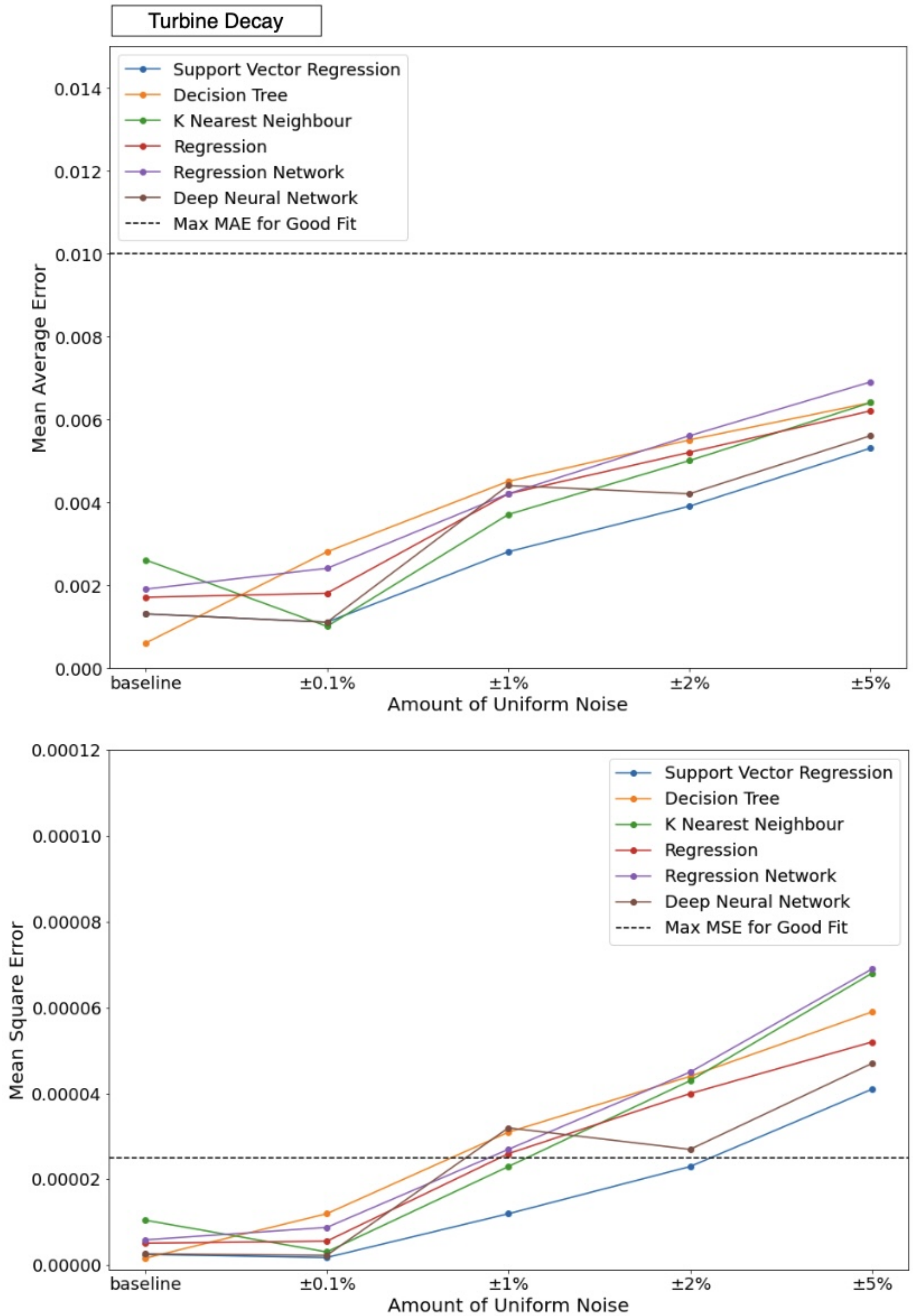

Figure 45: Turbine: mean average error and mean square error of the algorithms with various amounts of uniform error.

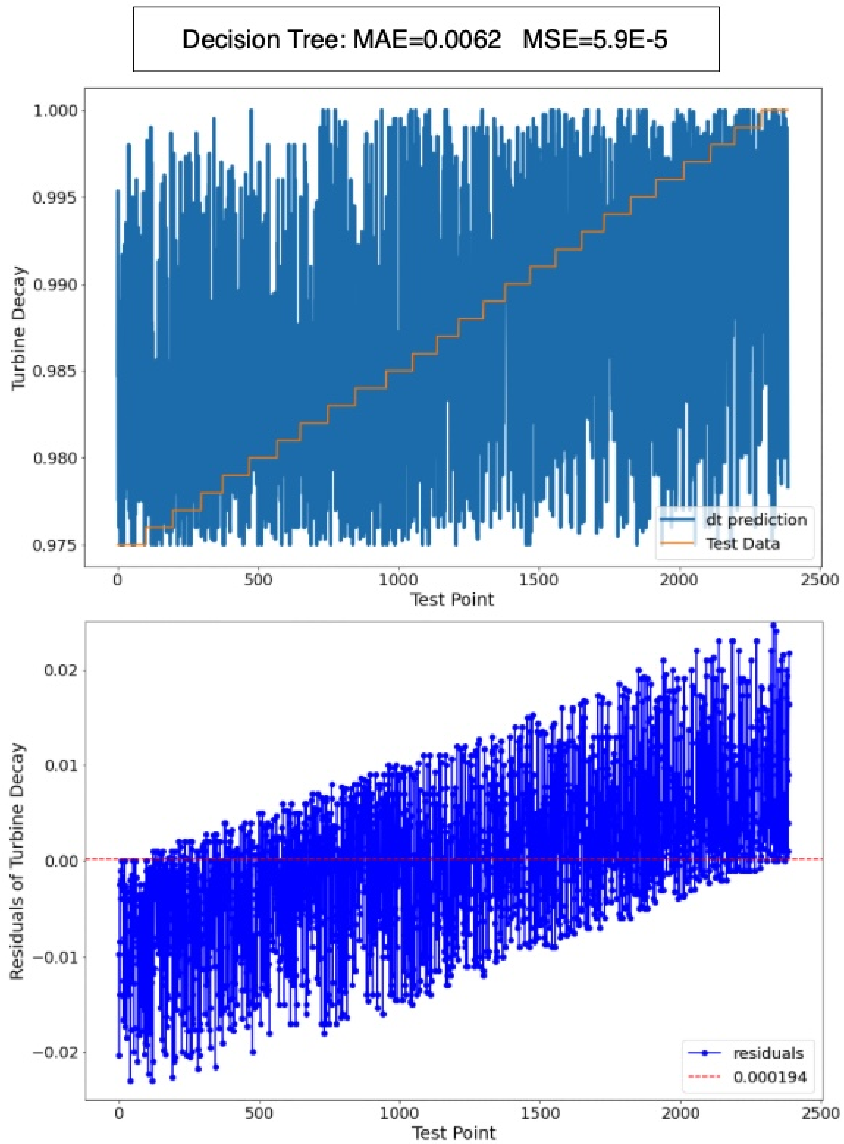

Figure 46: An example of a bad fit for uniform noise distribution ±5%; Decision Tree algorithm.

5.2.5 Selective Noise Results

The next set of experiments were performed by adding noise selectively to the input data. This was completed by taking a range of noise and randomly adding it to a selective sub-set of the input. The results can be seen in Table 8 and Figure 47 and 48. As seen in the figures, any amount of noise does increase the average error.

For the compressor decay prediction, the overall prediction ability was much lower than that of the turbine decay predictions with selective noise. This aligns with results from the previous experiment where, at around a 1% noise inclusion, the prediction abilities would fail based on the established criteria. The algorithms on the turbine decay prediction had better success.

The typical prediction ability can be seen in Figure 49, with the predictions for the DNN and the SVR models on the turbine data with noise of $\pm 5\%$ on 25% of the data. The predictions were approaching a bad fit as similar to the ones in Figure 46.

One outlier where a low MAE and MSE were still a bad fit can be seen in Figure 50 for the KNN fit on $\pm 1\%$ on 10% noisy data. Though the metrics are acceptable based on prior discussion of the MAE and MSE, the behaviours of the KNN on the beginning and end of the range do not make it a good fit.

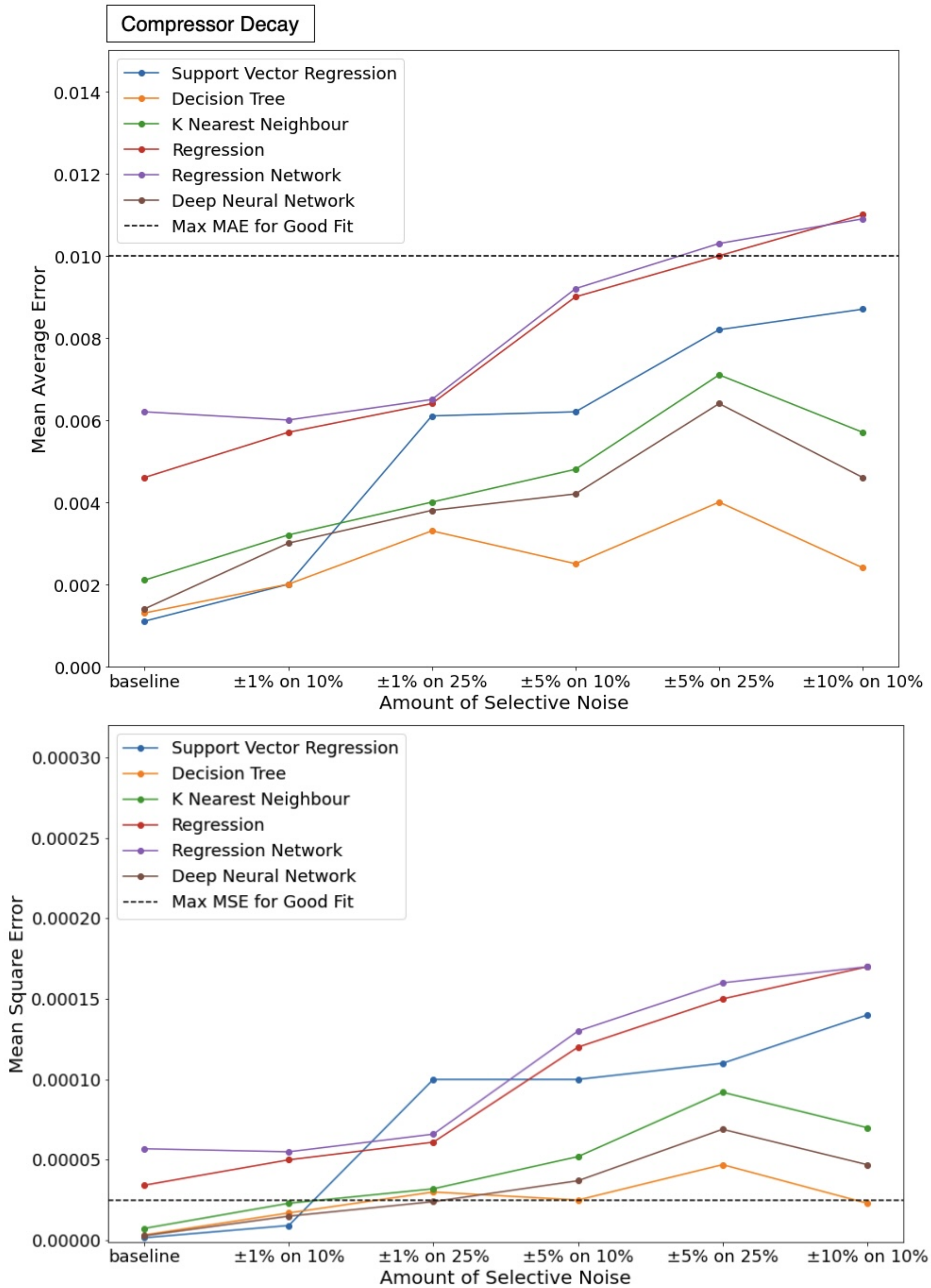

Figure 47: Compressor: Mean average error and mean square error of the algorithms with various amounts of selective error.

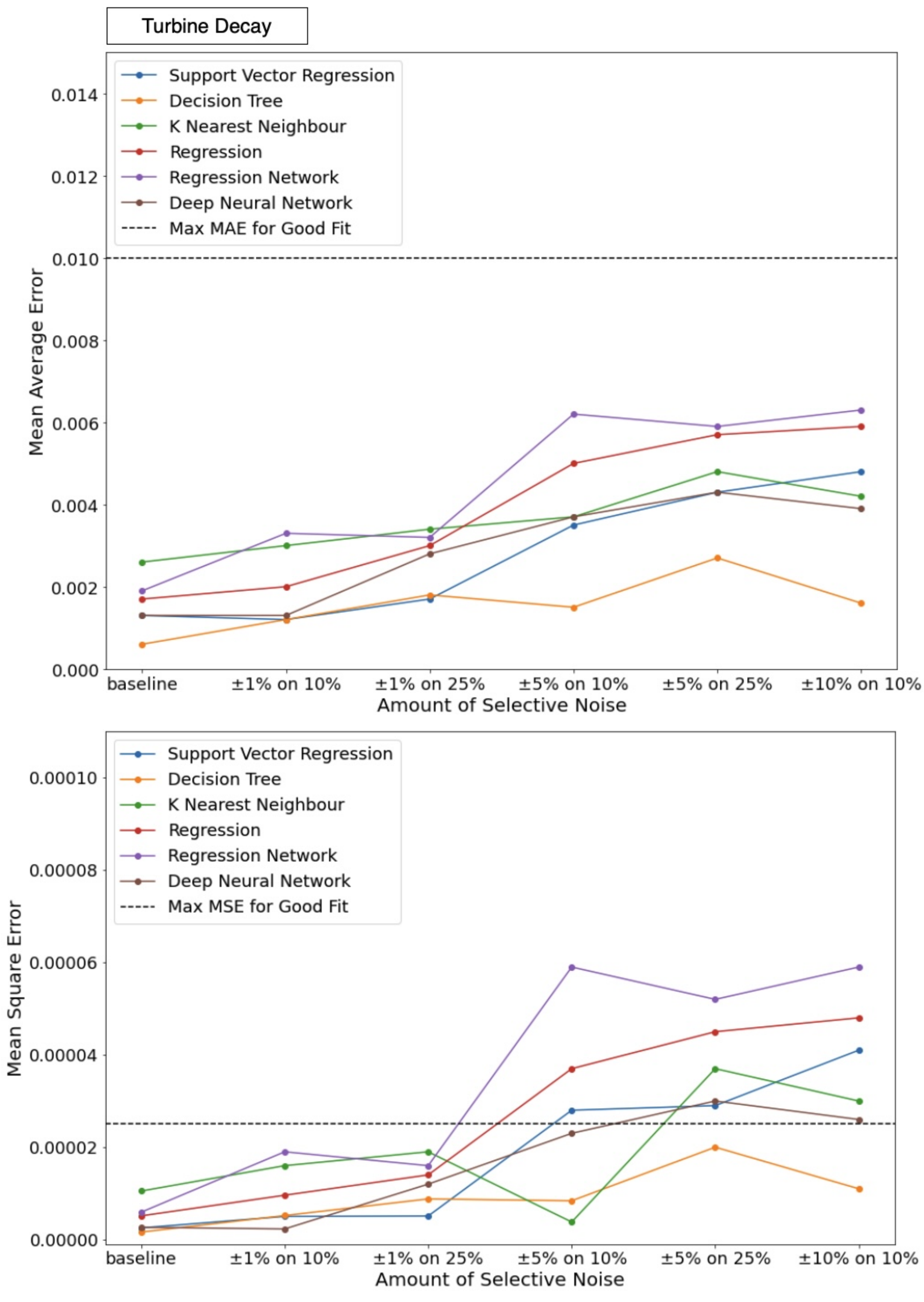

Figure 48: Turbine: Mean average error and mean square error of the algorithms with various amounts of selective error.

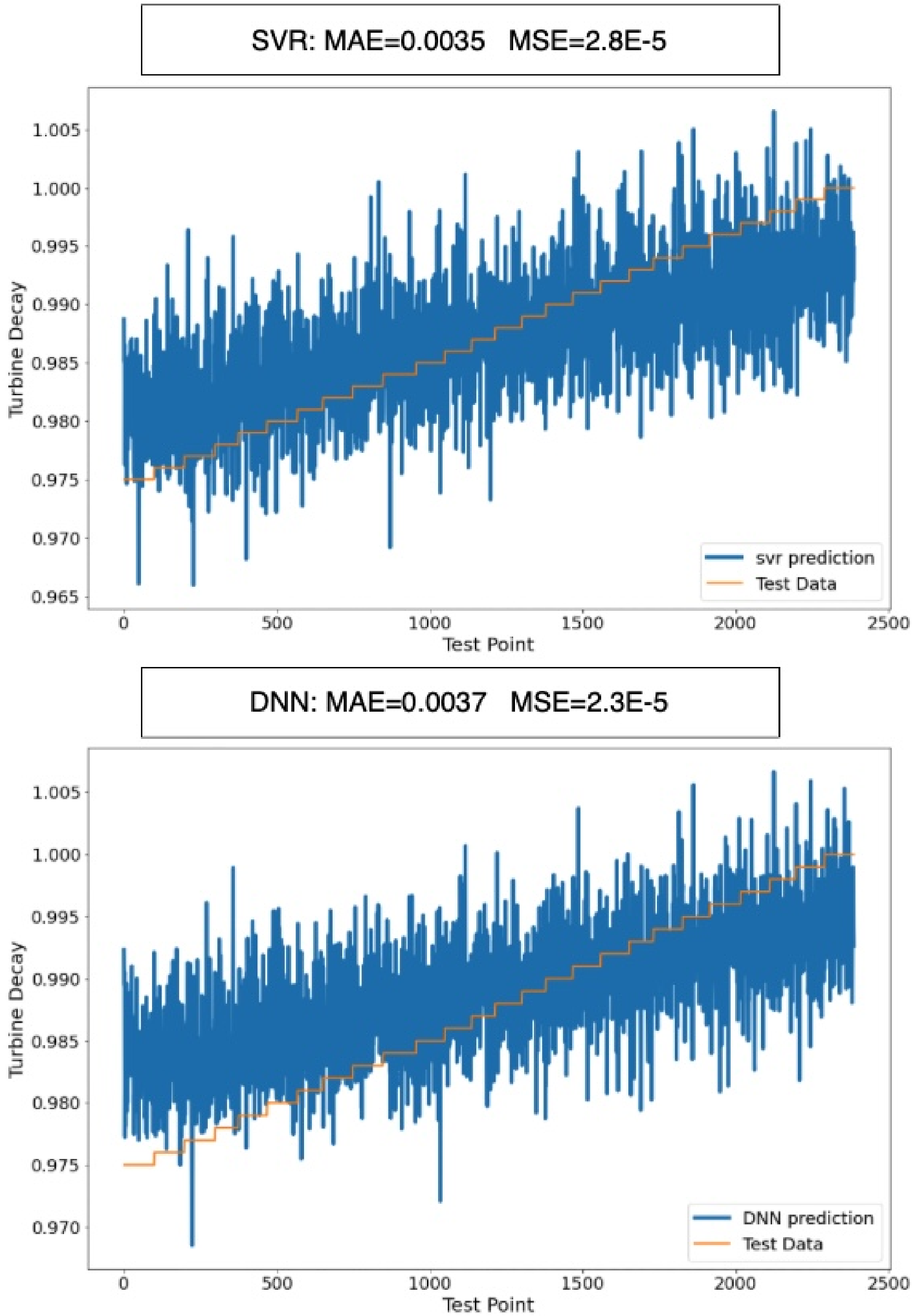

Figure 49: An example of an average fit for selective noisy data: ±5% noise on 25% of the input data.

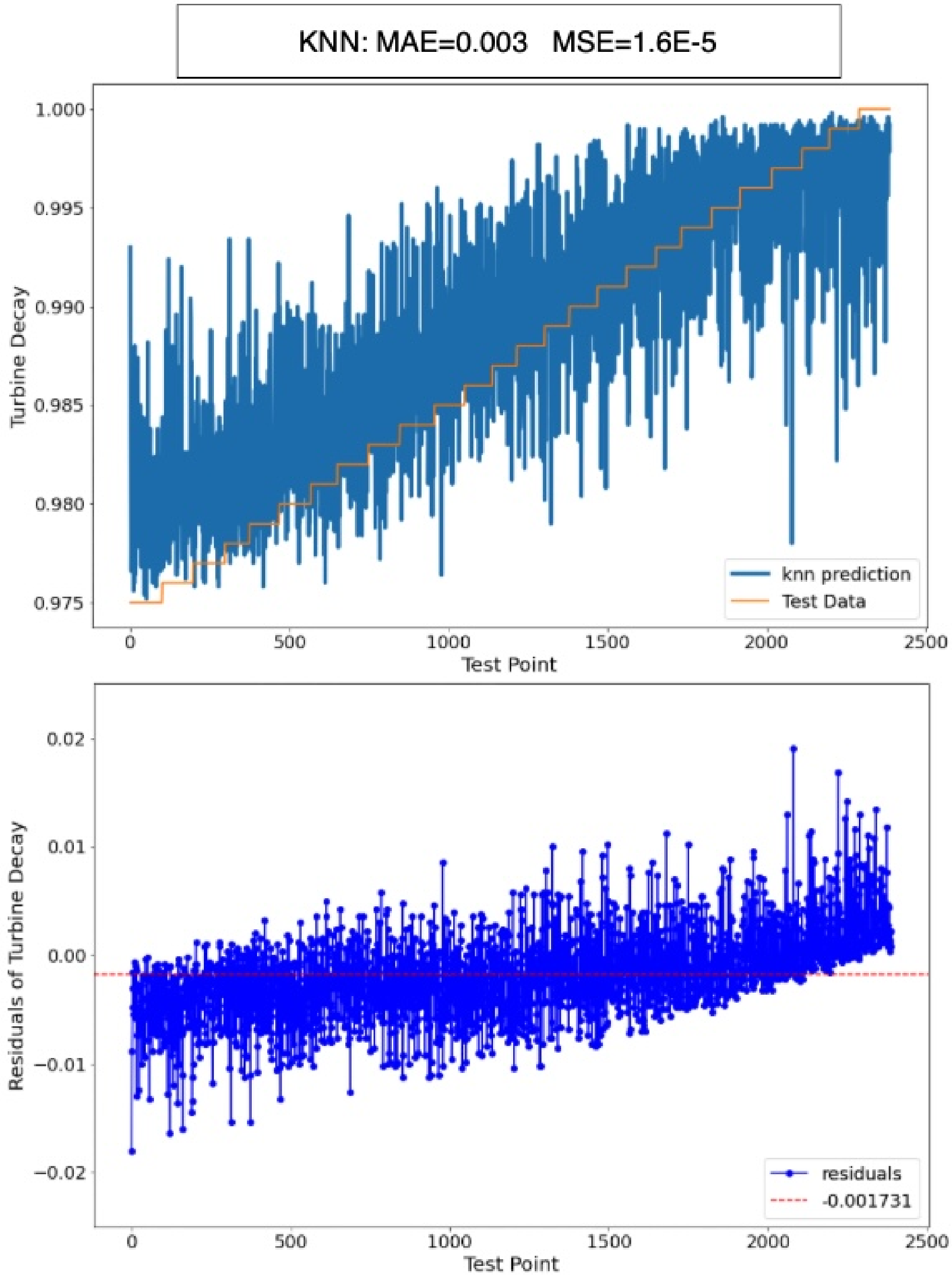

Figure 50: K Nearest neighbour fit of selective noisy data (±1% on 10%) for turbine decay prediction.

5.2.6 PCA Reduction

The next set of experiments was running the algorithms by reducing the dimensionality. The method selected was Principal Component Analysis (PCA). These sets of experiments were conducted to determine potential expected behaviour by removing attributes. Referring to Figure 51, 7 principal components made up for almost 100% of the variance in the input data. There first two principal components make up for 99% of the total variance however, as it can be seen in the results, additional principal components do aid in the prediction ability.

As seen in Figures 52 and 53, the more principal components, the closer the MAE gets to the baseline. For both decay predictions, there was no fit for the OLS multiple regression until 4 principal components. The numerical results are presented in Table 9.

For the compressor decay prediction the results show that a small number of principal components in this application are not sufficient and the results are not in a desired state until 7 components for certain algorithms. Like the noise experiments, the MSE for the compressor decay took a drastic hit. As seen in Figure 54, the DNN with one principal component has a MSE of $2.7*10^{-4}$ and is bad fit. This is a common behaviour seen with the predictions on the compressor using PCA.

For the turbine decay prediction, a low number of principal components is viable for running the models based on the MAE. However, based on the table and seen in Figure 53, increasing the number of principal components improves the MSE to the point where the SVR model is potentially viable using half the dimensionality. As seen in Figure 55 and 56 the SVR and DNN models have tight residuals.

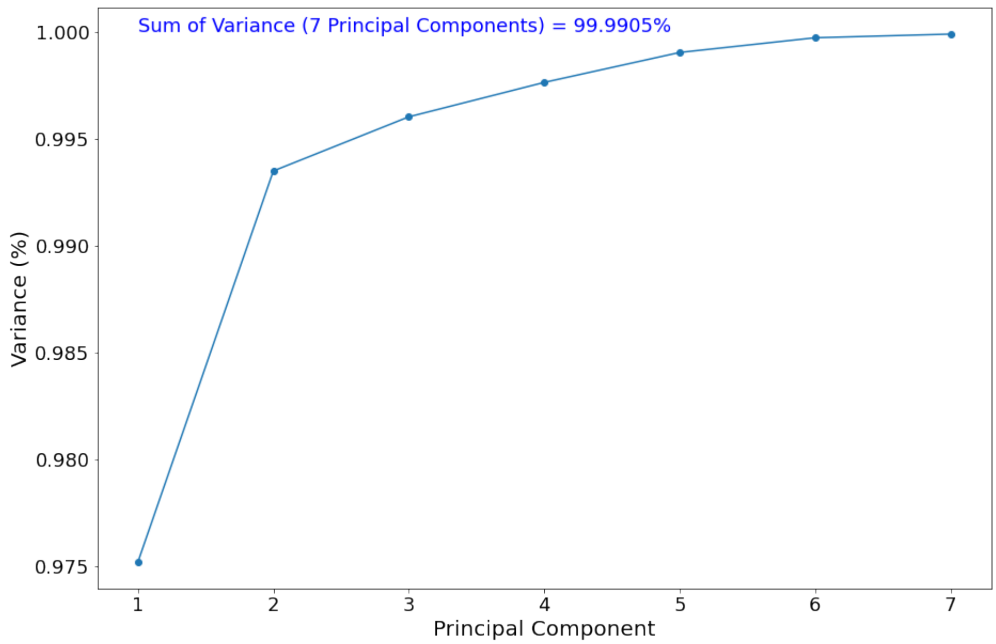

Figure 51: Scree plot to visualize principal component variance

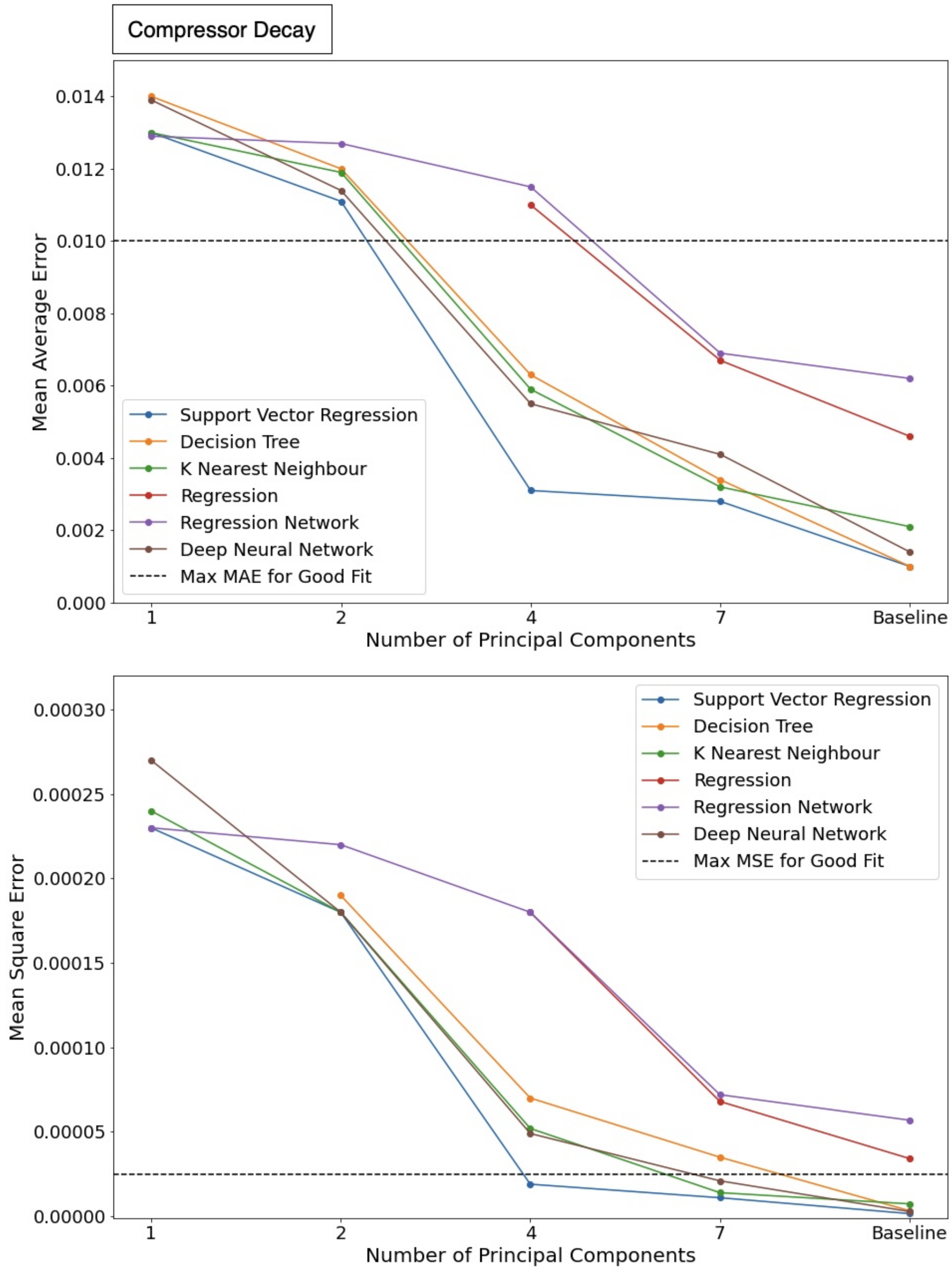

Figure 52: Compressor: Mean average error and mean square error of the algorithms with various principal components.

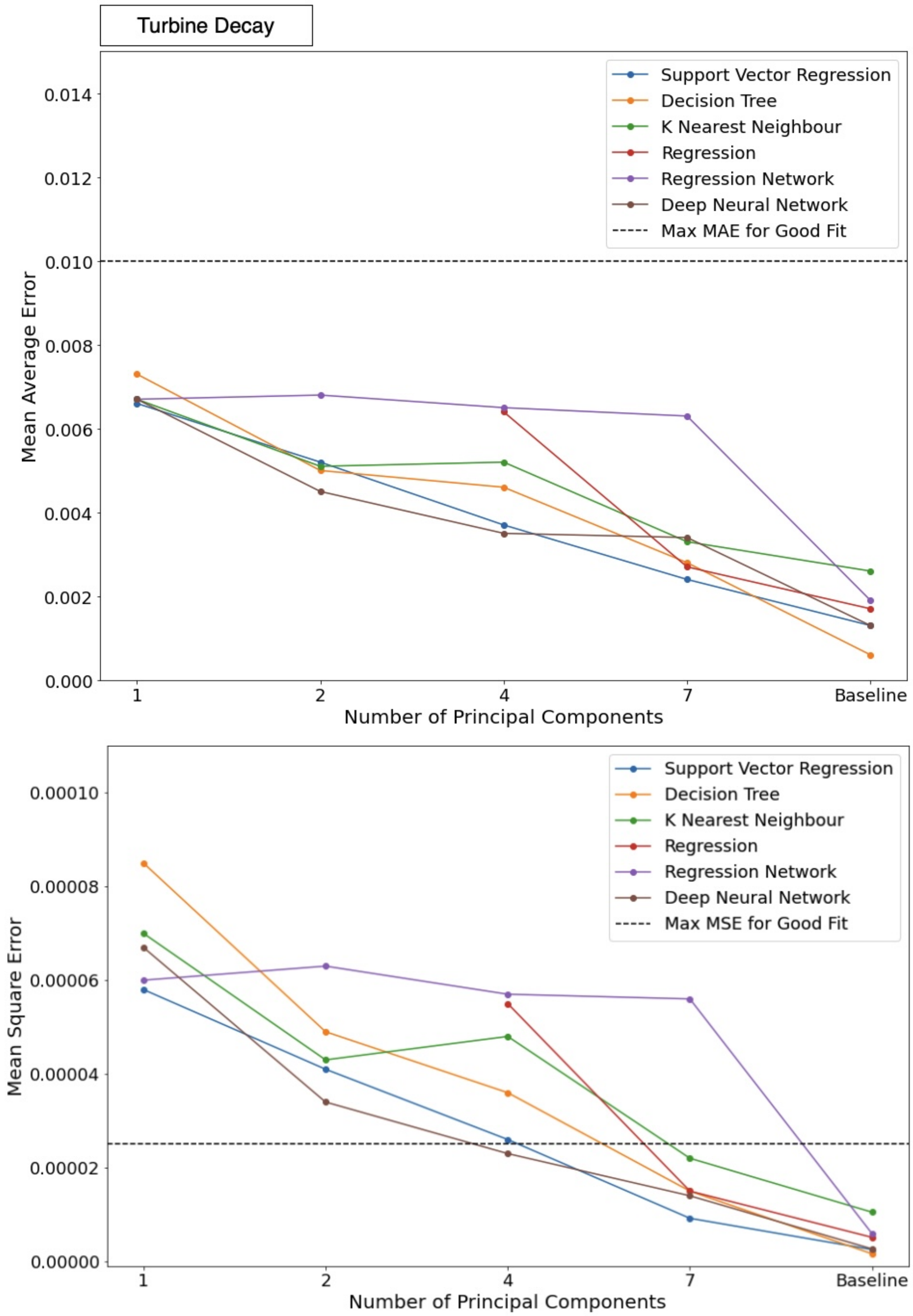

Figure 53: Turbine: Mean average error and mean square error of the algorithms with various principal components.

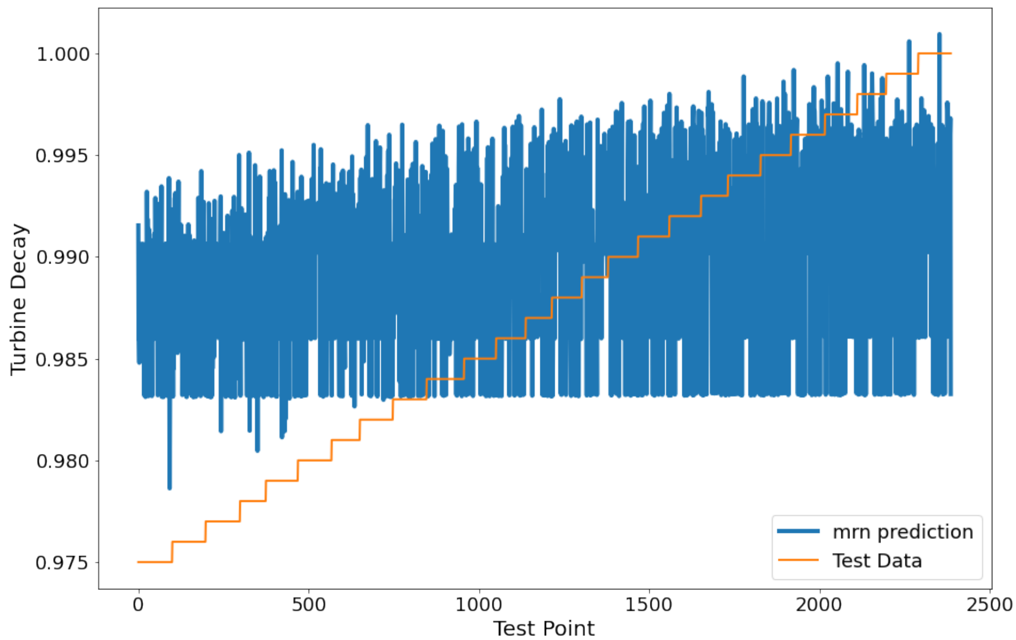

Figure 54: An example of a bad fit using 1 principal component trying to predict compressor decay using DNN.

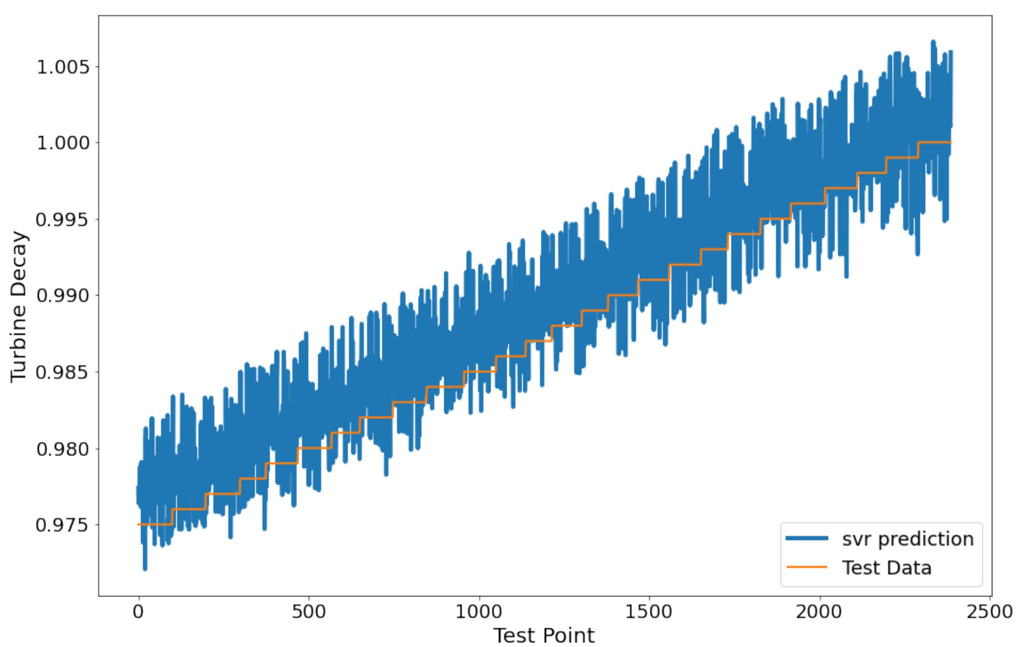

Figure 55: A good fit using the SVR trying to predict turbine decay using 7 principal components.

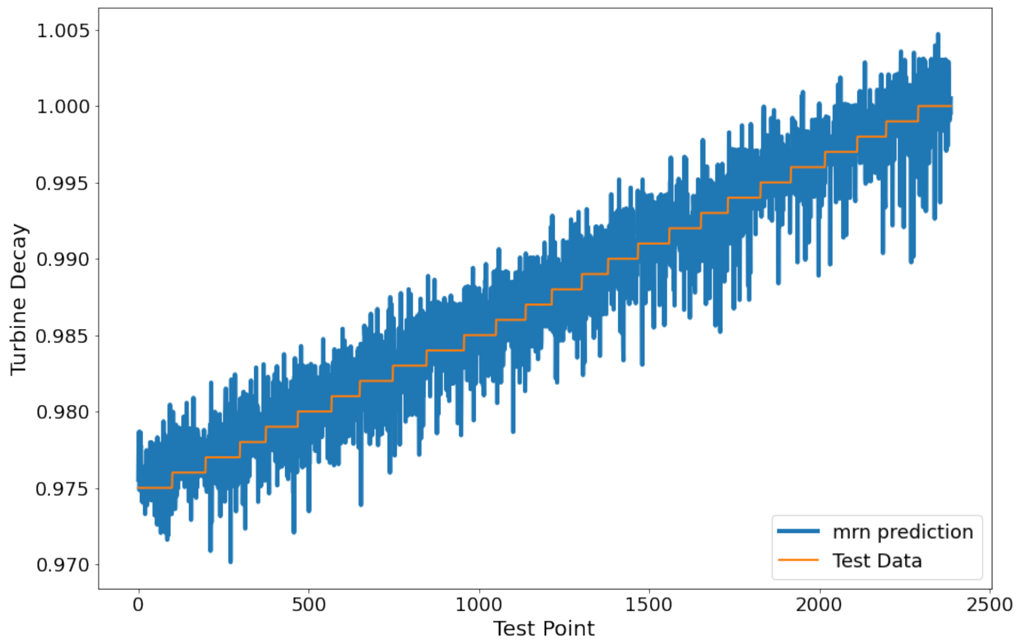

Figure 56: A good fit using the DNN trying to predict turbine decay using 7 principal components.

6 Discussion

The following section discusses how the results from the performed experiments align with the objectives of this study.

This section presents the proposed frameworks for maintenance delay prediction. First the two frameworks are discussed individually and then how they work together.

Two different frameworks are proposed for this work. The first framework allows for the prediction of maintenance delays directly based on historical maintenance logs. The second framework predicts the likelihood of delays based on functional and operational data of an asset and some sort of performance criteria similar to that of the data in this study.

Each of these frameworks have their benefits and downsides which will be discussed and the differences between the two serve as the contributions to this study.

6.1 Experimental Results

6.1.1 Prediction Score

As mentioned, a good prediction for this data is one where the MAE is lower than $1 * 10^{-2}$ and where the MSE is lower than $2.5 * 10^{-5}$. The MAE value was determined using the lowest reporting value of the sensor data. The values in the data set went to 3 decimal places which is a tenth of a percent of the actual decay values. The MSE value determined by plotting the prediction values against the labels and visually interpreting the fit as well as using the R^2 value. A threshold of 0.85 was set for the coefficient of determination to be considered a good fit. It was found that for this R^2 value, the trend for the associated MSE values were $2.5 * 10^{-5}$, thus defining the upper limit of the bound.

6.1.2 Best Algorithm

Upon establishing the baseline results, it was determined that for this data, the best results were found using the Support Vector Regression (SVR), Decision Tree Regression and the Deep Neural Network (DNN). The Multiple Regression using

Ordinary least squares was only successful with the turbine data and not the compressor data. This behaviour is postulated to be due to the lack of training cycles in the compressor decay. The turbine data had 52 maintenance cycles whereas the compressor has only 1. The K Nearest Neighbour (KNN) is not recommended for either prediction values because of the behaviour closer to the end of the cycle. As mentioned, the KNN averages the closest points when scanning across the domain of the prediction value. However, when the cycle of maintenance resets, there is a sudden shift from high reporting values to low reporting values, as seen in Figure 31. This periodic behaviour causes the algorithm to bias towards a lower value when trying to predict a higher value. This is not desired in this application because from a safety perspective, it is more important to be able to predict values representing asset failure to prevent corrective maintenance.

6.1.3 Noise Sensitivity

The inclusion of noise in small uniform amounts on only the test set was found to be a non-issue with the prediction ability however, with $\pm 1\%$ additional noise or more, the prediction ability fell. It is postulated that the poor performance with noisy data is due to over-fitting as the algorithms were not trained with such variety in input data. This can especially be seen with the Deep Neural Network. even though it performed the best in the baseline results, the high parameter count of the model caused over-fitting, making it the worst algorithm at higher noise levels.

Like the noise only on the test set, at $\pm 1\%$ noise, the algorithms started to perform worse. However, it is noted that in comparison to the previous experiments, the performance did not degrade by the same extent meaning that training with noise does aid in preventing over fitting. For the compressor prediction, only the KNN model was acceptable however, as mentioned in the baseline, the under-reporting the high values is still prevalent. For the turbine decay, the support vector regression was the best performing of all the models and was able to be in an acceptable prediction criteria with even $\pm 2\%$ uniform noise. At this range of noise, the neural network exceeded the maximum MSE score, however, it is by a negligible amount.

For the selectively added noise, similar to the uniform noise, the algorithms perform better in predicting the turbine decay than the compressor decay. Both regression methods failed in every experiment and are not recommended for compressor decay predictions. However, for the turbine decay, all the algorithms successfully ran with the ±1% noise on both 10% and 25% of the data. This aligns with the uniform noise where most algorithms were able to predict successfully under 100% of the data getting noise. The turbine prediction with selective noise shows that prediction ability is more sensitive by the degree of noise in comparison to the amount of noise. The increase in error between ±1%, ±5% and ±10% noise on 10% of the data is more substantial than the ±1% and ±5% noise increasing from 10% to 25% of the data set.

Based on these experiments, both the amount and degree of noise affects the prediction abilities. However, in low quantities, both in range and affected data, the prediction abilities are not affected. These set of experiments also present evidence that it is more important to minimize the range of noise than the affected sub-set of the data as well as training the algorithms with some noise included in the test set as it makes it more resilient to overfitting.

6.1.4 Dimensionality Sensitivity

Tests were performed to test the sensitivity of the prediction ability of the algorithms by reducing their dimensionality. The method chosen for this work reduction by Principal Component Analysis (PCA). This method was selected as it is commonly used to reduce high-dimensional sets of data into smaller, more interpretive sets. Based on the scree plot in Figure 51, the first 2 principal components account for 99% of the variance in the data. However, it was found using just the scree plot to determine how many principal components are enough to harbour good results is not enough. It was not until 7 Principal Components for both the turbine and compressor decay predictions until the reduction was successful.

However, at 7 principal components, the data set was reduced in half. This means that at 7 principal components and higher, the prediction capabilities are representative to the baseline results. This behaviour is important because in scenarios where feature selection needs to performed (where you gather the inputs), PCA is a good way of determining which attributes are of value and which do not provide any added benefit.

6.2 Framework 1: Maintenance Schedule Task Delay Prediction

Figure 57 presents the flowchart for Framework 1: Maintenance Schedule Task Delay Prediction. This framework provides a method on how to leverage historical information from maintenance logs for training data to predict potential delays in future maintenance schedules.

The framework is broken up into the following sections:

1. Historical Data Collection
2. Prediction Model Development
3. Delay Prediction & Schedule Improvement

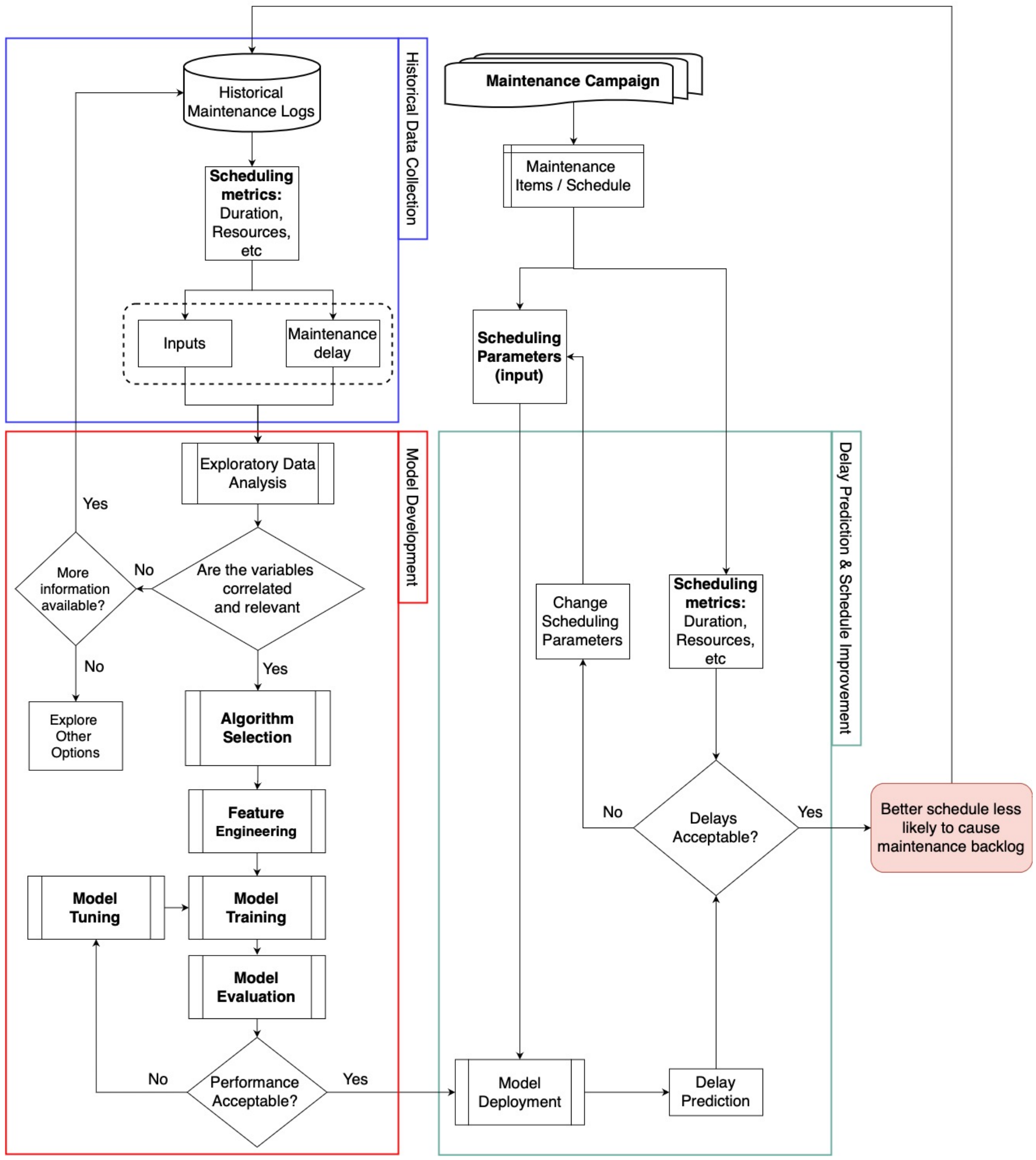

Figure 57: The flowchart for Framework 1: Scheduled Task Delay Prediction.

6.2.1 Historical Data Collection

As mentioned, many places such as nuclear power plants are rich with historical maintenance logs. The information from there can be extracted to serve as training data for the model.

The following are the proposed inputs of the proposed framework and how they were derived.

In Mofokeng's work on maintenance delays for the airline industry some of the causes for delays include the following [92]:

- Poor work planning,
- No adherence to work schedule,
- Delay in reporting conditions or status of maintenance,
- Asset condition worse than expected,
- Human resource restrictions, and
- Component unavailability.

Delay Analysis in Energy Utility Maintenance Project, cites contributing factors to delays in the coal-fired power plants [93]. These factors are extrapolated from a variety of studies from maintenance delay studies in construction, road maintenance, aircraft maintenance and nuclear power plants. The following are some of the more pertinent factors [93]:

- Deficiencies in initial negotiations,
- Material procurement,
- Inaccurate budgeting and resources,
- Design errors,
- Communication errors,
- Continuous change to requirements,
- Risk management,
- Quality management and monitoring,
- Low Productivity,
- Poor project management and project control,

- Excessive workload, and
- Errors in execution.

These studies conclude that each industry has unique mechanisms for the occurrence of their delays however the reasoning are similar and can be generically categorized.

Classification of Delay Factors:

As mentioned above, there are many reasons as to why a delay can occur during maintenance. However, it is possible to categorize the various factors. The categories are presented in Table 4.

Table 4: Delay Factor Classification

Factor Categorization	Example
Human Resources	Availability and accessibility for skilled workers to perform task
Procurement	Required components not available for execution
Project Management	Low productivity or unrealistic timeline
Execution Errors	Work load more than expected or emergent safety conditions
Technical Errors	Maintenance conditions different than expected

These categories align with the aforementioned literature specific to the nuclear industry. This allows for the creation of the proposed nuclear specific inputs for the proposed model framework. Based on delay prediction models on other industries such as the ones found in [58, 59, 60], Table 5 presents categories that were used to determine the specific input data.

Table 5: Model Input Categories

Input Category	Description
Project Identification	Most delay prediction models have identification that helps determine if individual proponents in the data are responsible for the delays
Initial Planning	In order to gauge the true delay value for every instance of data, the baseline projected metrics are needed. These initial planning metrics are dependant on the application and are usually accompanied by *true* value attributes
Execution	These are the true values of the task execution. These correspond to the initial planning attributes. Execution data and initial planning is also used to compute deltas. For example, true execution and planned execution time are used to determine the training data labels. The model is projected to be able to predict the delay based on initial planning. Thus these execution inputs are defined as planning factors right before execution.
Miscellaneous	These are application specific attributes that serve to provide the model with additional context or constraints that limit the model's predictions.

Maintenance Delay Data:

By using the model input categories and delay factor classifications specific to nuclear power plants, application specific inputs can be extrapolated. The following are the proposed inputs for a maintenance delay prediction that were developed. The following are the **Project Identification** inputs.

1. Maintenance task id: Every organization has specific method for classifying

their maintenance tasks. These are either coded ordinarily or using various alpha-numeric methods.

2. Project id: various maintenance projects involve multiple maintenance tasks and different projects may include the same maintenance task. An identification is needed to distinguish which project a task was scheduled in.

The following are the **Initial Planning** inputs.

1. Date: This represents the date a maintenance task is initially scheduled for.

2. Duration: This is the projected duration of a maintenance task. These are usually determined by based on data and technical expertise.

3. Location: Industrial facilities have complex geometries and the location helps provide context on the potential level of effort required. This may help the model determine if work in certain locations affect the execution time.

4. Degree of Maintenance: The total number of labour hours required for the task, including the set up, work execution, radiation protection etc.

5. Human Resource Requirement: Represents the amount of skilled workers to perform the maintenance task. For nuclear power plants this will also included the number of radiation protection personnel.

6. Type of Work: This attribute is categorical and represents the specific trade for the maintenance task. For example, electrical, structural, pie fitting etc.

7. Existing Backlog Level: This data provides contextual information to determine a if particular period has a higher backlog than normal.

The following are the **Execution** inputs.

1. Condition Upon Execution: It is common practice to perform condition checks prior to maintenance checks. These condition checks also confirm whether the Degree of Maintenance is correct.

2. Human Resource Availability: This attribute compares whether the initially planned required skilled workers are actually available or not.

3. Procurement Status: This confirms whether the required tooling and materials are completely ready, on route to be ready for execution, delayed or delayed until after projected execution period.

The following are the **Miscellaneous** inputs.

1. Frequency: This represents how often a maintenance task occurs during an assets life cycle.

2. Shutdown Level: Specifies to what extent an asset needs to shut off to. From no isolation to system wide shutdown.

3. Maintenance Period: This attribute is categorical which distinguishes which maintenance period a tasks occur in.

Figure 58 A figure of the proposed input data for Framework 1: Maintenance Schedule Task Prediction

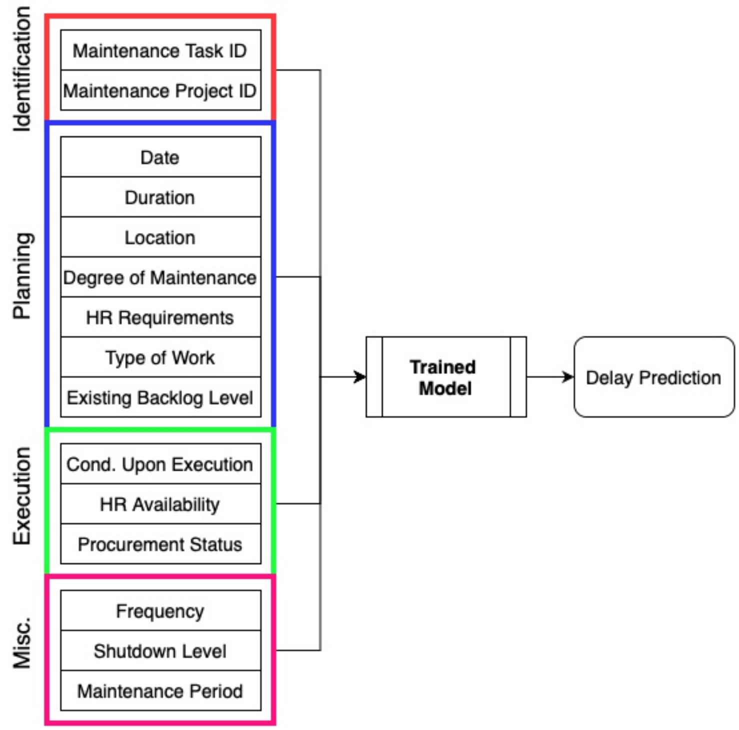

Figure 58: A list of inputs for maintenance task delay prediction based on framework 2.

As these inputs are proposed, all of these inputs are relevant to the problem. This is where methods such as Principal Component Analysis can aid in developing the prediction model using the proposed algorithms.

In this proposed method, the target variable is the delay time itself in minutes, hours etc.

6.2.2 Prediction Model Development

As shown in the work completed, the prediction model is developed through a process of first collecting the training data with the associated labels. Then exploratory data analysis is performed to understand the relations between the various variables.

Once the data is selected, many of the inputs are going to require some sort of feature engineering. Some examples include OneHotEncoding to vectorize categorical data or some sort of fourier transformation on cyclical information. This feature engineering is input dependent.

Once the data is in a format that can be used in various algorithms, there is a cycle of selecting the algorithm, training the model and then testing it to evaluate its performance. If the performance is not at an acceptable level, then there is a process of tuning the algorithm and re-training & testing. The acceptable level for error can be derived similar to the Mean Absolute and Mean Square Errors in this study.

6.2.3 Delay Prediction & Schedule Improvement

Once the model provides a delay prediction, it is compared to the initial planned duration of the task. Upon this comparison, there are 3 possibilities, the first being that the predicted delay is close to 0 within error, it is significantly over 0 or significantly under 0.

For predictions close to 0, it indicates that the task is likely to be completed on time. For tasks that are predicted to be significantly under 0, this means it is likely to be completed ahead of schedule. For values significantly over 0, the task is likely to be delayed based on the current scheduling parameters.

The definition of a *significant* delay is also organization dependent. Some potential methods include defining a significant delay based on the percentage of the original duration or having a fixed amount of time. This definition is important because it relates to the potential down-stream deferred maintenance effects. For example, a higher tolerance in delays would tell decision makers that their schedule is fine and to proceed as is. However, if multiple subsequent tasks have the same high allowable tolerance, tasks will be scheduled for longer than needed and as a result there will be lost potential of work and incur more costs for non-value added tasks. The resources are not being utilized to the best of their ability. On the contrary, if the definition of a significant delay is too restricted, there is no room for error or unexpected situations during the execution phase which could also lead to downstream backlog. If multiple tasks with a narrow definition of a significant delay are scheduled together, eventually the delays will add up and cause maintenance congestion resulting in more backlogged tasks.

Once a delay is defined, decision makers and planners can now go back into the original schedule and adjust the planning parameters and iterate on the schedule to improve it. Some examples include increasing the number of skilled workers on a task or adjusting the start date if there are conflicting tasks. This iterative process can continue until the decision makers and planners decide the schedule is in a better, desirable state than the original. As mentioned in Section 2.6.4, any number and combination of those KPIs can be used to evaluate the schedule, prior to and after execution.

Upon, completion of the improved schedule, the scheduling parameters and associated true delay times can then be fed back into the historical maintenance log and be used as more information to train the algorithms and the improvements made to the schedule can serve as *lessons learned* for future schedules.

6.3 Framework 2: Maintenance Delay Prediction Using Condition Data

Figure 59, presents the flowchart for Framework 2: Maintenance Delay Prediction Using Condition Data. This framework provides a method on how to take functional data such as the one found in the turbine & compressor decay prediction and mapping them to a performance metric that indicates the level of maintenance required.

The framework is broken up into two distinct sections with the following stages:

1. Model Prediction
   (a) Identification
   (b) Label Development
   (c) Model Development
2. Decision Making
   (a) Model Deployment & Verification
   (b) Linking Performance Criteria to Scheduling Metrics
   (c) Delay Postulation & Scheduling Decision

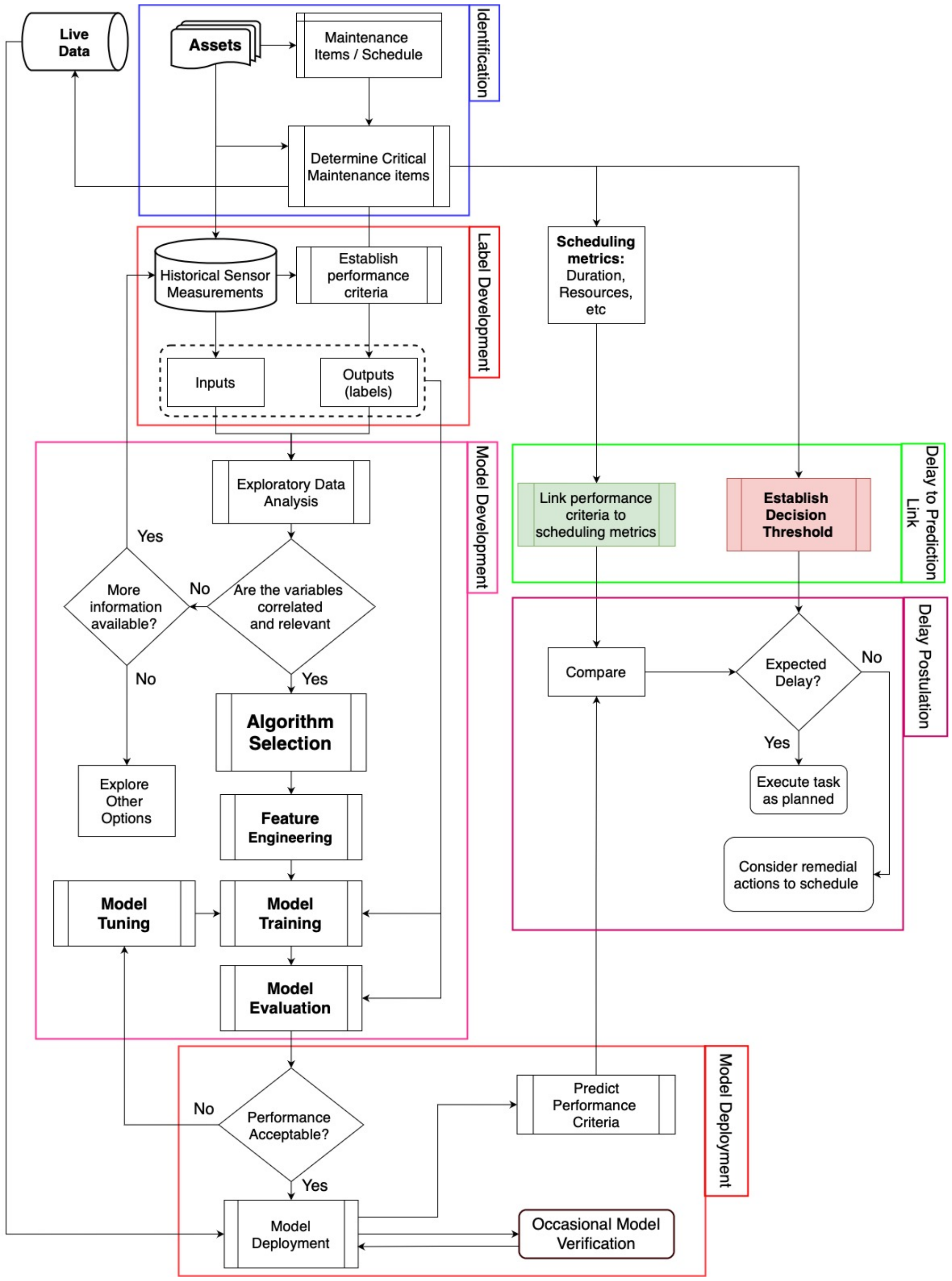

Figure 59: The flowchart for Framework 1: Performance Degradation to Delay Prediction.

6.3.1 Model Prediction

The first stage of the delay prediction framework is to be able to predict metric(s) related to maintenance requirement. This is done through a process of *Identification*, *Label Development*, and *Model Development*.

Identification: This stage of the framework establishes critical components and maintenance items. Based on the number of assets and maintenance items in a given maintenance campaign, and the limited amount of resources, not every task can be considered critical. Thus, to reduce downstream affects such as deferred maintenance, it is important to establish which critical maintenance tasks would incur substantial losses (based on the organization's objectives) if delayed.

As mentioned in the literature review, each organization has different objectives for their assets for which they perform maintenance. Some include meeting safety objectives, cost requirements, functional objectives etc. Cost analyses, safety analyses and quality control studies would establish the critical maintenance task.

Label Development: For any given asset the target variable (label) needs to be determined. In many scenarios the target is already established, in other cases they need to be formulated. For example, in the case of the turbine decay predictions, the target variable was established as the ratio of restricted gas flow to uninterrupted gas flow through the Gas turbine System. This label development should directly relate function to maintenance demand. As the decay coefficient increases, the flow of combustion gasses through the turbine and compressor is restricted and is a direct indication of required maintenance. In this work, the more restricted the gas flow is the more work (maintenance) is required to clean and open the flow. The target variable is often one that cannot be deterministically linked to the sensor information such as the correlations seen in Figure 33. If there is a deterministic approach to map the input features to a target variable, data driven models such as the ones presented in this work are not recommended as the deterministic method would always have a better accuracy than interpolation methods.

The next step is to gather all relevant historical information about the asset. This includes the historical sensor data as well as the scheduling metrics for the maintenance task. The scheduling metrics are to be used to relate the physical quantities of an asset to maintenance planning.

Model Development: Once the appropriate input training data and the associated labels are determined, the prediction can be developed. To determine the correct inputs, a process such as PCA can be used to extract the relevant information for the algorithms. This is an iterative process of selecting algorithms, training them and then testing them. In order to assess the algorithm, a performance criteria needs to be established. For numerical predictions such as the one in this work, the reduction in error is the performance criteria. The criteria is based on how much uncertainty is tolerable for a prediction. This tolerance is asset and risk dependent as established by the organization. A sample of this was completed in this work.

A potential method for determining the performance criteria of the prediction can be seen in Figure 60. Here, the performance metric can be a measure such as the turbine decay coefficient. Historical inspection cycles over time provide an expectation on how the asset will decay. The graph also color codes regions which represent the probability of failure depending on the performance of the asset. These same regions can also represent different levels of maintenance where, the dark green is *No Maintenance Required* and the dark red is *Replace Asset*. As the asset reaches closer to a fully decayed state, the probability of a failure event increases. Thus, the model's performance can either be adaptive or conservative. Adaptive such that, when the prediction is in a region of low failure probability, the margin for error in the prediction model can be greater as there is a higher risk tolerance and lower margin of error in more risk prone regions. The conservative approach would take the region of highest risk, determine its range as percent of the total and that becomes the maximum error tolerance for the prediction. For example, in Figure 60, the dark red region has the smallest range of 5%, thus, throughout the entire prediction range, the model should be able to predict within 5% of the true value.

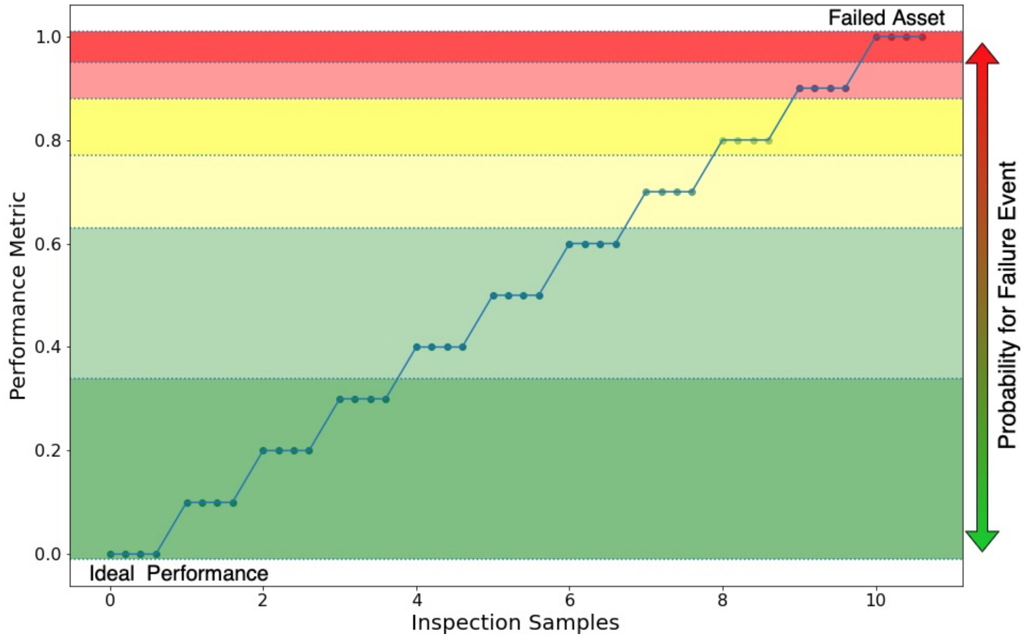

Figure 60: An example of mapping out the performance criteria of an asset to different risk categories.

6.3.2 Decision Making

The second part of the framework involves using the predicted values for decision making. In this stage of the framework, the output of the model prediction and the metric development are used to make an analytical decision.

Model Deployment & Verification Once the models are developed and their performance is acceptable, the model can be deployed. At this stage, continuous live data from the sensors can feed into the trained model to predict the live performance criteria. However to ensure that the results are continuously within the expected error ranges, the results would be verified through inspections of the asset and the results would be compared to the prediction values.

As shown through the experiments with the noise, the prediction performance of the models are inversely dependent on the amount of noise. As such, training with some artificial noise can make the prediction ability resilient to certain levels of noise.

However, the results showed at certain thresholds, the noise will make the predictions unusable. As such, this step in the framework is meant to ensure the data is of good quality.

Linking of Performance Criteria to Scheduling Metrics This stage of the framework is where performance of an asset is related to delays. Once the performance criteria of an asset is developed, it can be linked to the scheduling metrics. Using Figure 60 as an example, every color coded region is associated to a different work package with an increasing degree of maintenance required the closer the asset is to failure. In this example, cumulative labour hours among all workers is used as a metric for degree of maintenance. In the figure, the dark green region would require the lowest degree of maintenance as it is close to ideal performance and the dark red is the highest degree of maintenance as it is close to a failure event that would result in asset replacement.

Delay Postulation The prediction of the target is compared to the expected value. If the prediction value is within the error range of the expected value, then the degree of maintenance is as expected and the maintenance task is unlikely to be delayed. If the prediction is less than that of the expected value by a considerable margin, this may indicate a lower degree of maintenance required. If the prediction value is more, then this indicates the condition of the asset is degraded at a level higher than expected and a higher degree of maintenance is required. Once the delay is postulated, the decision makers can then try to find mitigating methods to reduce the potential delay. Examples of this include allocating more skilled workers to the task.

6.4 Comparison Between Framework 1 and Framework 2

6.4.1 Data Sourcing

The biggest difference in the proposed frameworks is the data and where they are sourced from. The data required to train the prediction models and the data for deploying these models differs as well. As mentioned in the previous section, when trying to predict delays directly from schedules as described in Framework 1 (Maintenance Schedule Task Delay Prediction), the training data comes from historical schedule maintenance logs whereas condition information for Framework 2 (Maintenance Delay Prediction Using Condition Data) comes sensor logs. Most maintenance schedules and logs are develop and recording using human input. However condition logs are information that comes directly from sensors.

For the deployment data, Framework 1 relies on maintenance planning schedules which are still currently developed manually. Framework 2 takes information directly from the asset. When addressing the sensitivity of the prediction models with respect to noise, the source differs as well. Maintenance schedules are more likely to have noise or errors due to human error whereas condition data comes directly from sensors and may be attributed to hardware issues.

This difference changes how the frameworks are implemented as sensor logs are more likely to be digitized where as historical maintenance logs may be on paper format or scanned.

6.4.2 Target Prediction

Framework 1 directly predicts maintenance delays where as Framework 2 predicts some sort of component performance variable that is defined in the data sourcing. With Framework 1, because the objective variable is directly being predicted, it makes the feature collection stage easier. When comparing Figures 57 and 59, the condition based Framework (2) has an additional stage of *Label Development* as well as *Delay to Prediction Link* because there is an additional step required to link condition based information to delays.

6.4.3 Model Development

The model development including training and testing is the same for both frameworks and the work previously outlined serves as an example of how this is conducted. The only difference between the two frameworks is the approach of the *Feature Engineering*. In this work and most condition based applications, the features of the models tend to be purely numerical in nature. However, as seen in Figure 58, many of the inputs for the direct delay prediction framework (1) are non-numerical in nature thus require different approaches to transform them into a format for the algorithms.

6.4.4 Deployment

For Framework 1, the model deployment is directly tied in with *Delay Prediction and Schedule Improvement* whereas Framework 2 has a specified *Model Deployment* stage. This is because the condition based model is a continuous model that uses information directly from sensors. This requires some continuous model verification because predicting prognostics of a component requires reliability on the system performing the prediction as well.

6.4.5 Metrics and Schedule Improvement

Both frameworks can use the same set of metrics such as mean-absolute error and mean-square error for developing the prediction model. However, the set of metrics each framework produces for assessing maintenance schedules is different. Predicting delays directly from a schedule can allow for the use of key performance indicators such as the ones shown in Section 2.6.4 wheres as the condition based approach requires each component to have their own performance criteria as shown in Figure 60.

This affects the usability of these frameworks because Framework 1 would allow for a standardized approach to assessing schedules across a group of maintenance tasks. This can be seen in Figure 57 with the output of the *Delay Prediction and Schedule*

Improvement feeding back directly into the global schedule. On the contrary, since the condition based approach to delays looks at singular components at a time, the *Delay Postulation* step in Figure 59 does not feed back into a global schedule. This is due to the indirect but focused approach to worrying only about critical components.

6.5 Framework 1 & 2 Interoperability

The purpose of having the two different frameworks is so that the they can work in conjunction with each other and aid in better decision making to reduce delayed maintenance tasks which subsequently reduces deferred maintenance.

As mentioned in the literature review, Time Based Maintenance (TBM) is still the predominant maintenance approach in nuclear power plants. Over the years, more condition information has been implemented to improve the TBM approach but there is still room to improve it. Framework 1 (Maintenance Schedule Task Delay Prediction), establishes a method to be able to predict delays given a schedule but does not consider any condition information, it is still a postulated duration of execution based strictly on historical experience. For the critical components in a given maintenance campaign, condition based information such as the work shown with the turbine decay prediction is needed as an additional factor of consideration, hence the development of Framework 2 (Maintenance Delay Prediction Using Condition Data). This is due to the fact that many assets in a nuclear power plant are purpose built and unique to the application. Many of these components may only go under maintenance a handful of times in a power plant's life cycle thus the use of framework 1 may incur a higher uncertainty than a component that regularly undergoes maintenance due to a lack of training data. In this scenario, framework 2 was proposed so that the delay of an asset can be determined using its performance rather than planning metrics.

The way that both these approaches would work in tandem is to assess regularly occurring tasks through framework 1 and then critical components using framework 2. The decisions to then alter and improve the schedule would rely on the delay perceived from critical components as a priority. As the delays for the critical components are known through their condition, the tasks surrounding or reliant on the resources from the critical tasks can be rescheduled and their potential delays can be determined using framework 1.

The reason for a two framework approach is to be able to select whichever configuration is suitable for the application. Either of the frameworks can be applied individually and the use of them together is proposed to harbour the most benefit. The way the two frameworks will work with each other can be seen in Figure 61. It is to be noted that both *Framework 1 & Framework 2* processes in Figure 61 assume the appropriate models have trained algorithms where as Figures 57 and 59 shows the entire process to get the models trained and deployed.

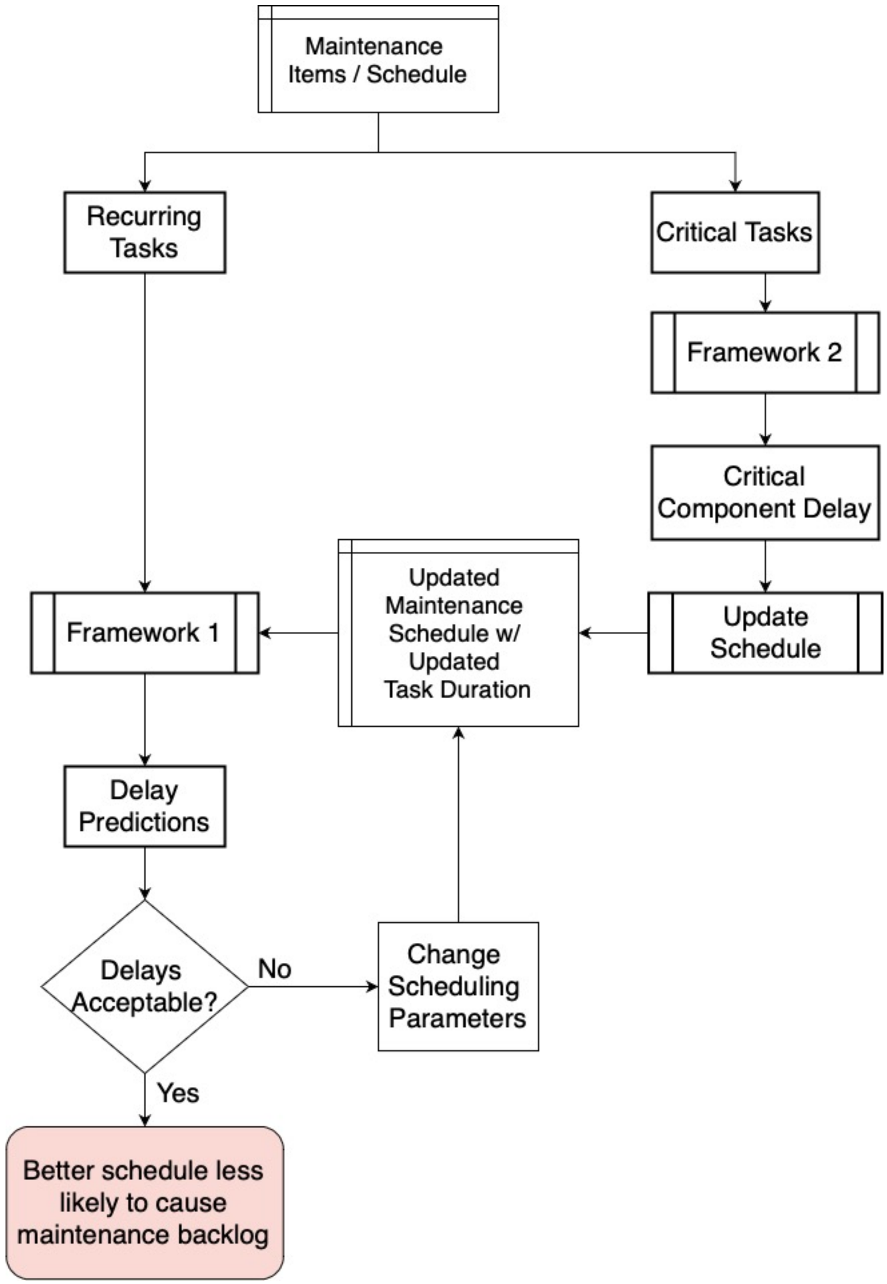

Figure 61: Interoperability of the 2 separate proposed frameworks.

6.6 Fulfillment of Thesis Objectives

The outlined objectives for this thesis as described in the introduction of this work can be said to have been fulfilled. The objectives were as follows:

1. Investigate how maintenance is planned in nuclear power plants.

2. Identify nuclear power plant-specific maintenance planning considerations.

3. Correlate maintenance delays to deferred maintenance and their consequences.

4. Explore and test data-driven methods to predict a target variable and how it can be implemented for the use case of maintenance task delays.

5. Develop a proof of concept framework(s) to predict maintenance delays and identify how decision makers and maintenance planners can use it to analyze maintenance schedules.

Objectives 1,2 and 3 were completed by the included literature review to discuss how maintenance planning practices can lead to maintenance delays. Then a link was established between maintenance delays and deferral of tasks into backlog due to cascading downstream affects of unattended delays.

Objective 4 was completed through the various experiments where many different approaches for supervised machine learning were performed. This included some sensitivity analysis with noise and changing the dimensionality.

Objective 5 was completed by developing two different frameworks for maintenance delay predictions, the first predicting delays directly using the postulated input data and a second that postulates a delay through condition based information. The use case and the differences for each of the frameworks were discussed as well as how they would work together. These framework also discussed where the data is sourced, and the flow of data through various stages of the frameworks.

6.7 Limitations & Future Work

The following are some of the identified limitations with the current approach and associated proposed future work.

The first limitation of this work is the range of the prediction value. The compressor decay was sampled between the values of [0.950:1.000] and the turbine decay was between [0.975:1.000]. The behaviour of the current algorithms may have different behaviour for ranges from [0.000:1.000]. Also, for the compressor decay, there was only a single decay cycle. Future work would involve running the algorithms through multiple decay cycles.

In addition to the range within the target variable, another limitation is that the data used was based on simulator data for an application, though similar, not the intended application. As such, future work would include testing the algorithms and repeating the experiments with real Class III Power diesel turbine system from a nuclear power plant.

In this work, the various numerical experiments were used to help develop the frameworks. Future work would include testing these frameworks and determining whether it is applicable for the intended use.

Also, based on Figure 31, it is noted that the decay patterns are behaving linearly, thus the rate of decay is constant and more predictable. However, different components and systems may have degradation patterns that are non-linear. A next step for this research is to sample assets that have varying patterns of decay.

The previous limitations mentioned were ones that can be addressed in the near term. However, there are other limitations that require a lot more extensive work.

One of the major limitations of this work is not being able to assess the algorithms for use in Framework 1 using the postulated data that was identified. The next stage of this research would involve sourcing the appropriate desired data and trying to predict the delays. The data postulated for use in this framework is based on historical maintenance logs, thus future work would involve sourcing them, determining how to process the data, testing the algorithms and the framework.

As mentioned, one of the limitations in nuclear data is the lack of historical failure events and their associated data. In this work, the turbine had 52 cycles in which the decay coefficient went all the way to 1 (failure). In the nuclear power plant perspective, a turbine would not operate to a decay coefficient of 1, thus there is a lack of data. Future work would include determine methods to either develop the algorithms to predict the failure events without the data or develop ways to generate this data.

Additionally, when considering the application for nuclear power plant assets, the abundance of data and the operational time serves to be a limitation as well. The current data set was around 12000 instances which is relatively small compared to the data. Thus, in training of the model, future work would include creating a methodology on how to obtain the data, ensuring it is ideal for training and determining whether sensor drift and other factors have impacted the data fidelity.

In Framework 2, a major component of the framework is *Live Data*. However, based on the nuclear data considerations mentioned in the literature review, to test and evaluate the framework, additional work is required to determine the sampling methodology and the frequency of the data for prediction. This would then affect how sensor verification is performed to ensure there is not an unacceptable amount of noise in the data.

This work was completed with a target variable (decay coefficient) already developed. There was a single metric that could represent the condition of the turbine and compressor, however, not all assets may have a metric like this developed. Future work would include developing a methodology on determining if it is feasible to represent asset condition based on a singular metric and how to derive these target variables. Additionally, the framework requires a link between the target variable such as the decay coefficient and the maintenance level of effort and scheduling parameters. Future work would include deriving the metrics between condition and scheduling data.

7 Concluding Remarks

This work served as a feasibility study on the usage of predictive analytics to predict maintenance delays. The concluding remarks of this work are summarized in the following:

- Various supervised learning algorithms were tested using a representative dataset that presents the behaviours of a gas-turbine system similar to ones that can be found in nuclear power plants. The algorithms were used to predict *Turbine Decay* and *Compressor Decay Coefficients* as the target variables.

- The Support Vector Regression (SVR) and the Deep Neural Network (DNN) had the best prediction ability among the 6 different algorithms tested. The SVR and DNN algorithms had a Mean Absolute Error of 0.0011 and 0.0014 respectively for the Compressor Decay prediction and 0.0013 and 0.0013 respectively for Turbine Decay prediction. The Mean Square Error was 1.60E-6 and 2.83E-6 respectively for the Compressor Decay prediction and 2.52E-6 and 2.66E-6 respectively for the Turbine Decay prediction. In this study a suitable prediction was established as one where the Mean Average Error and Mean Square Error are lower than 1E-2 and 2.5E-5 respectively .

- It was found that training these algorithms using clean data and then deploying them with the use of noisy data is an area of concern as the prediction ability decreases severely. A method to improve prediction ability with the use of noisy data is by training the algorithms with artificially induced noise. It was found by including uniformly distributed noise in the training set, the algorithms were able to have suitable predictions for noise ranges of $\pm 1\%$ and below. For selectively distributed noise, it was found that there is a trade off between the amount of affected data and the extent of the noise. A small affected set of the data with a high level of noise can reduce the prediction ability just as much as a low range of noise affecting a larger subset of the data. Thus, for the use in nuclear power plants, it is recommended to train the algorithms with artificially added noise so that the models have some resilience to certain levels of incorrect data.

- By reducing the dimensionality of the data using Principal Component Analysis, it was found that for this data set, the prediction ability of the Deep Neural Network and the Support Vector Regression was still suitable at half of the dimensionality (7 principal components). For the application in nuclear power plants, using PCA is helpful in trying to remove low value features and scaling the prediction ability to additional features.

- One framework (Maintenance Schedule Task Delay Prediction) was proposed to predict maintenance delays directly from maintenance logs. The input data for this framework was proposed and future work is required to obtain and test the algorithms. The benefit of this framework is that it directly predicts delays and has a cyclical stage in which a maintenance schedule can be altered and tested to improve the schedule.

- A second framework (Maintenance Delay Prediction Using Condition Data) was proposed to use condition monitoring data such as the one found in the data set to predict the state of a component and then postulate the associated maintenance level. In this framework, critical assets are considered and a target variable is defined that is associated with the health of the asset. The framework uses the proposed algorithms to predict the target variable and if the prediction differs from the expected value, there is a disparity in the expected maintenance and true maintenance required. The benefit of this framework is that it considers the prognostics of critical components and establishes maintenance decisions based on numerical performance metrics such as decay coefficients. The drawback of this framework is that it requires having a direct association between the target variable and the level of maintenance required.

- It was discussed how the two frameworks compliment one another. The framework using the condition data can establish the maintenance level required for critical assets and the framework directly predicting the maintenance delays can be used to help plan and reduce delays in other non critical maintenance tasks. The combination of these frameworks and the use of machine learning can aid stakeholders in determining potential adverse conditions of components earlier when compared to traditional inspection methods and reduce maintenance activities that are scheduled but not required. This in turn prevents maintenance delays and subsequent maintenance deferrals which reduces costs assumed by the nuclear power plant and protects the overall health of the asset.

- The completed work presents a method on digitizing and modernizing an asepct of maintenance practices in nuclear power plants. The use of predictive analytics and advanced machine learning algorithms in the nuclear industry would help costs reductions and provide a better understanding of the maintenance operations. Small efficiencies gained by using these technologies in reducing maintenance delays would scale up tremendously in nuclear power plants as there are thousands of personnel and maintenance activities occurring which would result in massive cost and time savings.

References

[1] R. Ahmad and S. Kamaruddin, "An overview of time-based and condition-based maintenance in industrial application," *Computers and Industrial Engineering*, vol. 63, no. 1, pp. 135–149, 2012.

[2] S. Martorell, J. F. Villanueva, S. Carlos, Y. Nebot, A. Sánchez, J. L. Pitarch, and V. Serradell, "RAMS+C informed decision-making with application to multi-objective optimization of technical specifications and maintenance using genetic algorithms," *Reliability Engineering and System Safety*, vol. 87, no. 1, pp. 65–75, 2005.

[3] H. Rødseth and P. Schjølberg, "Maintenance backlog for improving integrated planning," *Journal of Quality in Maintenance Engineering*, vol. 23, no. 2, pp. 195–225, 2017.

[4] "Maintenance strategy design for nuclear reactors safety systems using a constraint particle swarm evolutionary methodology," *Annals of Nuclear Energy*, vol. 150, p. 107878, 2021.

[5] C. Lu, J. Lyu, L. Zhang, A. Gong, Y. Fan, J. Yan, and X. Li, "Nuclear power plants with artificial intelligence in industry 4.0 era: Top-level design and current applications—a systemic review," *IEEE Access*, vol. 8, pp. 194 315–194 332, 2020.

[6] G. Rothwell, *Economics of nuclear power.* Routledge, 2016.

[7] J. Galabert, D. Hopper, J. C. Neviere, D. Nodwell, R. Pascal, Y. Poitevin, I. Ricapito, and G. White, "Nuclear maintenance strategy and first steps for

preliminary maintenance plan of the EU HCLL & HCPB Test Blanket Systems," *Fusion Engineering and Design*, vol. 116, pp. 34–39, 2017.

[8] N. Paltrinieri, L. Comfort, and G. Reniers, "Learning about risk: Machine learning for risk assessment," *Safety Science*, vol. 118, no. May, pp. 475–486, 2019.

[9] J. Lim, H. Kim, and Y. Park, "Review of the regulatory periodic inspection system from the viewpoint of defense-in-depth in nuclear safety," *Nuclear Engineering and Technology*, vol. 50, no. 7, pp. 997–1005, 2018.

[10] L. Krishnasamy, F. Khan, and M. Haddara, "Development of a risk-based maintenance (RBM) strategy for a power-generating plant," *Journal of Loss Prevention in the Process Industries*, vol. 18, no. 2, pp. 69–81, 2005.

[11] A. K. S. J. Ã, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, pp. 1483–1510, 2006.

[12] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics : A systematic review from data acquisition to RUL prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018.

[13] H. Kupwade Patil and T. Chen, *Chapter 18. Wireless Sensor Network Security*, 2017.

[14] A. N. Das, F. L. Lewis, and D. O. Popa, "Data-logging and supervisory control in wireless sensor networks," in *Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06)*, 2006, pp. 330–338.

[15] B. Saha and D. Srivastava, "Data quality: The other face of big data," in *2014 IEEE 30th International Conference on Data Engineering*, 2014, pp. 1294–1297.

[16] 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, no. 1, pp. 437–446, 2015.

[17] B. de Jonge, R. Teunter, and T. Tinga, "The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance," *Reliability Engineering and System Safety*, vol. 158, no. August 2016, pp. 21–30, 2017.

[18] Z. Sheng, C. Mahapatra, C. Zhu, and V. C. Leung, "Recent advances in industrial wireless sensor networks toward efficient management in IoT," *IEEE Access*, vol. 3, no. Oma Dm, pp. 622–637, 2015.

[19] A. Tripathi, H. P. Gupta, T. Dutta, R. Mishra, K. K. Shukla, and S. Jit, "Coverage and Connectivity in WSNs: A Survey, Research Issues and Challenges," *IEEE Access*, vol. 6, pp. 26 971–26 992, 2018.

[20] A. Chammas, M. Traore, E. Duviella, M. Sayed-Mouchaweh, and S. Lecoeuche, "Condition monitoring architecture for maintenance of dynamical systems with unknown failure modes," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 45, no. 31, pp. 36–41, 2012.

[21] W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans, "Trial without error: Towards safe reinforcement learning via human intervention," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2018, p. 2067–2069.

[22] "Optimal maintenance policy for a Markovian system under periodic inspection," *Reliability Engineering and System Safety*, vol. 71, no. 2, pp. 165–172, 2001.

[23] Fiix, "How to make condition-based maintenance more effective," 2019.

[24] R. M. Ayo-Imoru and A. C. Cilliers, “A survey of the state of condition-based maintenance (CBM) in the nuclear power industry,” *Annals of Nuclear Energy*, vol. 112, pp. 177–188, 2018.

[25] R. Krivanek, “Factors limiting lifetime of nuclear power plants with pressurized-water reactors,” *Nuclear Engineering and Design*, vol. 370, no. October, p. 110872, 2020.

[26] M. C. Leva, G. Baldissone, R. Caso, M. Demichela, L. Lawlor, and B. McAleer, “Cost benefit evaluation of maintenance options for aging equipment using monetised risk values: A practical application,” *Procedia Manufacturing*, vol. 19, no. 2017, pp. 119–126, 2018.

[27] S. H. D. Mandelli, C. Wang, M. Abdo, A. Alfonsi, P. Talbot, J. Cogliati, C. Smith, D. Morton, I. Popova, “Light Water Reactor Sustainability Program Development and Application of a Risk Analysis Toolkit for Plant Resources Optimization,” Idaho National Labratory, Tech. Rep. September, 2020.

[28] J. Pence, M. Abolhelm, Z. Mohaghegh, S. Reihani, M. Ertem, and E. Kee, “Methodology to evaluate the monetary benefit of Probabilistic Risk Assessment by modeling the net value of Risk-Informed Applications at nuclear power plants,” *Reliability Engineering and System Safety*, vol. 175, no. March 2016, pp. 171–182, 2018.

[29] I. Cambridge Systematics, “Consequences of Delayed Maintenance - NCHRP 14-20 - interim report,” Tech. Rep. September, 2011.

[30] A. S. Tam and J. W. Price, “A maintenance prioritisation approach to maximise return on investment subject to time and budget constraints,” *Journal of Quality in Maintenance Engineering*, vol. 14, no. 3, pp. 272–289, 2008.

[31] Z. Seif, E. Mardaneh, R. Loxton, and A. Lockwood, “Minimizing equipment shutdowns in oil and gas campaign maintenance,” *Journal of the Operational Research Society*, no. 0, pp. 1–19, 2020.

[32] X. Wang, Y. Zhang, L. Wang, J. Wang, and J. Lu, "Maintenance grouping optimization with system multi-level information based on BN lifetime prediction model," *Journal of Manufacturing Systems*, vol. 50, no. October 2018, pp. 201–211, 2019.

[33] G. Bosco, "Practical Methods for Optimizing Equipment Maintenance Strategies Using an Analytic Hierarchy Process and Prognostic Algorithms," *Masters Theses*, 2017.

[34] S. M. H. Hadavi, "Risk-Based, genetic algorithm approach to optimize outage maintenance schedule," *Annals of Nuclear Energy*, vol. 35, no. 4, pp. 601–609, 2008.

[35] C. N. S. Commission, "Refurbishment and life extension," 2015.

[36] R. Flin, P. O'Connor, and K. Mearns, "Crew resource management: Improving team work in high reliability industries," *Team Performance Management: An International Journal*, vol. 8, pp. 68–78, 2002.

[37] Canadian Nuclear Safety Commission, "Protecting workers," 2019.

[38] IAEA, "Procurement Engineering and Supply Chain Guidelines in Support of Operation and Maintenance of Nuclear Facilities," Tech. Rep., 2016.

[39] S. E. Fawcett, G. M. Magnan, and M. W. McCarter, "Benefits, barriers, and bridges to effective supply chain management," *Supply Chain Management*, vol. 13, no. 1, pp. 35–48, 2008.

[40] P. Sandborn, "Design for obsolescence risk management," *Procedia CIRP*, vol. 11, pp. 15–22, 2013.

[41] G. Doe and C. B. Ramsey, "Nuclear Facility Maintenance Management Program Guide for Use with DOE O 433 . 1B," pp. 6–14, 2011.

[42] P. Oedewald and N. Gotcheva, "Safety culture and subcontractor network governance in a complex safety critical project," *Reliability Engineering and System Safety*, vol. 141, pp. 106–114, 2015.

[43] P. Oedewald, N. Gotcheva, T. Reiman, E. Pietikäinen, and L. Macchi, "Managing safety in subcontractor networks: The case of olkiluoto 3 nuclear power plant construction project," in *Proceedings of the 4th Symposium on Resilience Engineering*, 2011, 4th Symposium on Resilience Engineering ; Conference date: 08-06-2011 Through 10-06-2011.

[44] C. N. S. C. (CNSC), *Fitness for Service Reliability Programs for Nuclear Power Plants*, 2017, no. August.

[45] "Plant life management and maintenance technologies for nuclear power plants," *Hitachi Review*, vol. 50, no. 3, pp. 84–88, 2001.

[46] Ontario Power Generation, "Ontario Power Generation Inc. Annual Information Form (for the year ended December 31, 2018)," 2019.

[47] I. A. E. Agency, "Application of Reliability Centred Maintenance to Optimize Operation and Maintenance in Nuclear Power Plants," International Atomic Energy Agency, Tech. Rep. IAEA-TECDOC-1590, 2007.

[48] I. N. Outage and G. Plan, "Production Forecast and Methodology – Nuclear," 2012.

[49] P. Contri, I. Kuzmina, and B. Elsing, "Maintenance optimization and nuclear power plant life management - A proposal for an integrated set of maintenance effectiveness indicators," *Journal of Pressure Vessel Technology, Transactions of the ASME*, vol. 134, no. 3, pp. 1–6, 2012.

[50] S. Alsharif and A. Karatas, "A Framework for Identifying Causal Factors of Delay in Nuclear Power Plant Projects," *Procedia Engineering*, vol. 145, no. 248, pp. 1486–1492, 2016.

[51] N. Alp and B. Stack, "Scope management and change control process study for project-based companies in the construction and engineering industries," *2012 Proceedings of Portland International Center for Management of Engineering and Technology: Technology Management for Emerging Technologies, PICMET'12*, pp. 2427–2436, 2012.

[52] Ontario Power Generation, "Overview of nuclear facilities EB-2007-0905," Tech. Rep., 2007.

[53] B. McGee, "2017 Business Plan," Pickering, 2017.

[54] H. Pham and H. Wang, "Imperfect maintenance," *European Journal of Operational Research*, vol. 94, no. 3, pp. 425–438, 1996.

[55] S. Kannan, Y. Li, N. Ahmed, and Z. El-Akkad, "Developing a Maintenance Value Stream map," *Institute of Industrial Engineers, Technical Societies and Divisions Lean Conference Proceedings*, pp. 1–8, 2007.

[56] J. Chen and Y. Wang, "A resource demand prediction method based on EEMD in cloud computing," *Procedia Computer Science*, vol. 131, pp. 116–123, 2018.

[57] M. Verma, N. C. N. G. R. Gangadharan, R. Vadlamani, V. Inamdar, and Lakshmi Ramachandran, "Dynamic resource demand prediction and allocation in multi-tenant service clouds," *Concurrency Computation Practice and Experience*, vol. 22, no. 6, pp. 685–701, 2016.

[58] A. Q. Gbadamosi, L. O. Oyedele, J. M. D. Delgado, H. Kusimo, L. Akanbi, O. Olawale, and N. Muhammed-yakubu, "IoT for predictive assets monitoring and maintenance: An implementation strategy for the UK rail industry," *Automation in Construction*, vol. 122, no. August 2019, p. 103486, 2021.

[59] S. Ayvaz and K. Alpay, "Predictive Maintenance System for Production Lines in Manufacturing: A Machine Learning Approach Using IoT Data in Real-Time," *Expert Systems with Applications*, vol. 173, no. November 2020, p. 114598, 2021.

[60] D. Truong, “Using causal machine learning for predicting the risk of flight delays in air transportation,” *Journal of Air Transport Management*, vol. 91, no. November 2020, p. 101993, 2021.

[61] X. Wu, W. Zheng, X. Chen, Y. Zhao, T. Yu, and D. Mu, “Improving high-impact bug report prediction with combination of interactive machine learning and active learning,” *Information and Software Technology*, vol. 133, no. October 2020, p. 106530, 2021.

[62] H. Hashemian, *Nuclear Power Plant Instrumentation and Control*, 09 2011.

[63] U.S Department of Homeland Security, Cybersecurity and Infrastructure Security Agency, “Nuclear Sector: Cybersecurity Framework Implementation Guidance,” Tech. Rep., 2020.

[64] A. L. Fradkov, “Early History of Machine Learning,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1385–1390, 2020.

[65] O. Duerr, B. Sick, and E. Murina, *Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability.* Manning Publications, 2020.

[66] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

[67] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixão, F. Mutz, L. de Paula Veronese, T. Oliveira-Santos, and A. F. De Souza, “Self-driving cars: A survey,” vol. 165, no. August 2020, p. 113816, 2021.

[68] A. Kannagi, J. Gori Mohammed, S. Sabari Giri Murugan, and M. Varsha, “Intelligent mechanical systems and its applications on online fraud detection analysis using pattern recognition K-nearest neighbor algorithm for cloud security applications,” *Materials Today: Proceedings*, no. xxxx, 2021.

[69] P. Rajendiran and P. Priyadarsini, “Survival study on stock market prediction techniques using sentimental analysis,” *Materials Today: Proceedings*, no. xxxx, 2021.

[70] A. Amrouche, M. Debyeche, A. Taleb-Ahmed, J. Michel Rouvaen, and M. C. Yagoub, "An efficient speech recognition system in adverse conditions using the nonparametric regression," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 1, pp. 85–94, 2010.

[71] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg, "Activeclean: Interactive data cleaning for statistical modeling," *Proc. VLDB Endow.*, vol. 9, no. 12, p. 948–959, Aug. 2016.

[72] V. N. Gudivada, J. Ding, and A. Apon, "Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations," *International Journal on Advances in Software*, vol. 10, no. 1, pp. 1–20, 2017.

[73] M. J. Eppler and M. Helfert, "A classification and analysis of data quality costs," *Proceedings of the Ninth International Conference on Informationen (ICIQ-04)*, no. February, pp. 311–325, 2004.

[74] G. Seber and A. Lee, *Linear Regression Analysis*, ser. Wiley Series in Probability and Statistics. Wiley, 2012.

[75] N. Beck, "OLS in Matrix Form," *Lecture Notes*, vol. 1, no. 2, pp. 1–14, 2001.

[76] B. Wang and V. Pavlu, "Support Vector Machines," Tech. Rep., 2015.

[77] S.-W. Hsu, C.-C. Chnag, and C.-J. Lin, "A Practical Guide to Support Vector Classification," Tech. Rep., 2016.

[78] C. C. Chang and C. J. Lin, "LIBSVM: A Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–39, 2011.

[79] L. Rokach and O. Maimon, "Decision Trees," *Lecture Notes in Mathematics*, vol. 1928, pp. 67–86, 2008.

[80] A. N. Elmachtoub, J. C. N. Liang, and R. McNellis, "Decision trees for decision-making under the predict-then-optimize framework," *37th International Conference on Machine Learning, ICML 2020*, vol. PartF168147-4, pp. 2838–2847, 2020.

[81] C. J. S. Leo Breiman, Jerome H. Friedman, Richard A. Olshen, *Classification and Regression Trees*, 2011.

[82] C. C. M. Shalizi, "Regression : Predicting and Relating Quantitative Features," Tech. Rep., January 2011.

[83] L. F. O. L. Rosasco, "Lecture 2- Local Methods and Bias Variance Trade-Off K-Nearest Neighbor Parzen Windows High Dimensions," Tech. Rep., 2014.

[84] I. T. Jollife and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, 2016.

[85] J. Shlens, "A Tutorial on Principal Component Analysis," *Measurement*, pp. 1–13, 2005.

[86] F. Cipollini, L. Oneto, A. Coraddu, A. J. Murphy, and D. Anguita, "Condition-based maintenance of naval propulsion systems: Data analysis with minimal feedback," *Reliability Engineering and System Safety*, vol. 177, pp. 12–23, 2018.

[87] Eric A. Therrio, "Machine Learning Techniques For Development Of A Condition-Based Maintenance Program For Naval Propulsion Plants," Ph.D. dissertation, 2018.

[88] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, and M. Figari, "Machine learning approaches for improving condition-based maintenance of naval propulsion plants," *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, vol. 230, no. 1, pp. 136–153, 2016.

[89] Eric A. Therrio, "Machine Learning Techniques For Development Of A Condition-Based Maintenance Program For Naval Propulsion Plants," Ph.D. dissertation, 2018.

[90] T. Berghout, L.-H. Mouss, T. Bentrcia, E. Elbouchikhi, and M. Benbouzid, "A deep supervised learning approach for condition-based maintenance of naval propulsion systems," *Ocean Engineering*, vol. 221, p. 108525, 2021.

[91] P. Pal, R. Datta, A. Segev, A. Yasinsac *et al.*, "Condition based maintenance of turbine and compressor of a codlag naval propulsion system using deep neural network," in *6th international conference on artificial intelligence and applications (AIAP-2019)*, 2019.

[92] T. Mofokeng, "Factors contributing to delays of aircrafts during A-check maintenance," pp. 1 – 104, 2012.

[93] K. Adiam and H. Nel, "Delay analysis in energy utility maintenance project," *Proceedings of the International Conference on Industrial Engineering and Operations Management*, no. July, pp. 343–354, 2019.

Appendices

1 Test Only Noise

Table 6 presents the results of the algorithm runs with uniform noise only on the test set.

Table 6: Results of Algorithms with Uniform Noisy Data on Test Set Only

Noise	±0.1%		±1%		±2.5%		±5%	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Compressor Decay								
SVR	0.0015	3.23E-06	0.0150	0.0002	0.024	0.0009	0.0451	0.0031
DT	0.0062	0.0001	0.0140	0.0003	0.017	0.0005	0.0195	0.0006
KNN	0.0028	1.60E-06	0.0130	0.0002	0.014	0.0003	0.0172	0.0004
MR	0.0047	3.77E-05	0.0150	0.0004	0.035	0.0020	0.0687	0.0075
MR-N	0.0060	5.22E-05	0.0124	0.0003	0.028	0.0012	0.0521	0.0042
DNN	0.0021	7.00E-06	0.0161	0.0004	0.036	0.0020	0.0771	0.0088
Turbine Decay								
SVR	0.0014	3.10E-06	0.0061	5.73E-05	0.0141	0.0003	0.0260	0.0011
DT	0.0033	3.93E-05	0.0072	8.86E-05	0.0089	0.0001	0.0100	0.0014
KNN	0.0027	1.22E-05	0.0058	4.20E-05	0.0069	7.80E-05	0.0083	0.0014
MR	0.0019	5.97E-06	0.0074	7.58E-05	0.0151	0.0004	0.0274	0.0013
MR-N	0.0028	1.24E-05	0.0066	6.64E-05	0.0201	0.0007	0.0312	0.0016
DNN	0.0026	9.61E-06	0.0143	0.0006	0.0341	0.0021	0.0697	0.0087

2 Uniform Noise Results

Table 7 presents the results of the algorithm runs using the uniform noise distribution.

Table 7: Results of Algorithms with Uniform Noisy Data

Noise	±0.1%		±1%		±2%		±5%	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Compressor Decay								
SVR	0.0010	2.8E-06	0.0051	4.2E-05	0.0075	8.8E-05	0.0100	1.6E-04
DT	0.0019	7.6E-06	0.0065	7.8E-05	0.0090	1.5E-04	0.0120	3.0E-04
KNN	0.0026	1.1E-05	0.0067	4.5E-06	0.0090	1.3E-04	0.0114	2.0E-04
MR	0.0049	3.7E-05	0.0077	8.8E-05	0.0100	1.3E-04	0.0120	1.9E-04
MR-N	0.0051	4.0E-05	0.0079	9.2E-05	0.0114	1.2E-04	0.0117	1.9E-04
DNN	0.0017	4.4E-06	0.0052	4.2E-05	0.0074	8.9E-05	0.0105	1.6E-04
Turbine Decay								
SVR	0.0011	1.8E-06	0.0028	1.2E-05	0.0039	2.3E-05	0.0053	4.1E-05
DT	0.0028	1.2E-05	0.0045	3.1E-05	0.0055	4.4E-05	0.0064	5.9E-05
KNN	0.0010	3.1E-06	0.0037	2.3E-05	0.0050	4.3E-05	0.0064	6.8E-05
MR	0.0018	5.6E-06	0.0042	2.6E-05	0.0052	4.0E-05	0.0062	5.2E-05
MR-N	0.0024	8.8E-06	0.0042	2.7E-05	0.0056	4.5E-05	0.0069	6.9E-05
DNN	0.0011	2.3E-06	0.0044	3.2E-05	0.0042	2.7E-05	0.0056	4.7E-05

3 Selective Noise Results

Table 8 presents the results of the algorithm runs using the uniform noise distribution on partial parts of the data.

Table 8: Results of Algorithms with Selective Noisy Data

Noise (%)	±1		±1		±5		±5		±10	
Affected Data (%)	10		25		10		25		10	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Compressor Decay										
SVR	0.0020	9.2E-06	0.0061	1.0E-04	0.0062	1.0E-04	0.0082	1.1E-04	0.0087	1.4E-04
DT	0.0020	1.7E-05	0.0033	3.0E-05	0.0025	2.5E-05	0.0040	4.7E-05	0.0024	2.3E-05
KNN	0.0032	2.3E-05	0.0040	3.2E-05	0.0048	5.2E-05	0.0071	9.2E-05	0.0057	7.0E-05
MR	0.0057	5.0E-05	0.0064	6.1E-05	0.0090	1.2E-04	0.0100	1.5E-04	0.0110	1.7E-04
MR-N	0.0060	5.5E-05	0.0065	6.6E-05	0.0092	1.3E-04	0.0103	1.6E-04	0.0109	1.7E-04
DNN	0.0030	1.5E-05	0.0038	2.4E-05	0.0042	3.7E-05	0.0064	6.9E-05	0.0046	4.7E-05
Turbine Decay										
SVR	0.0012	5.01E-6	0.0017	5.1E-06	0.0035	2.8E-05.	0.0043	2.9E-05	0.0048	4.1E-05
DT	0.0012	5.2E-06	0.0018	8.8E-06	0.0015	8.4E-06.	0.0027	2.0E-05	0.0016	1.1E-05
KNN	0.0030	1.6E-05	0.0034	1.9E-05	0.0037	3.8E-06.	0.0048	3.7E-05	0.0042	3.0E-05
MR	0.0020	9.6E-06	0.0030	1.4E-05	0.0050	3.7E-05.	0.0057	4.5E-05	0.0059	4.8E-05
MR-N	0.0033	1.9E-05	0.0032	1.6E-05	0.0062	5.9E-05	0.0059	5.2E-05	0.0063	5.9E-05
DNN	0.0013	2.3E-06	0.0028	1.2E-05	0.0037	2.3E-05	0.0043	3.0E-05	0.0039	2.6E-05

4 Principal Component Analysis Results

Table 9 presents the results of the algorithm runs after conducting a principal component analysis .

Table 9: Results of Algorithms with PCA Reduction

Principal Components	1		2		5		7	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Compressor Decay								
SVR	0.0130	2.3E-04	0.0111	1.8E-04	0.0031	1.9E-05	0.0028	1.1E-05
DT	0.0140	3.0E-03	0.0120	1.9E-04	0.0063	7.0E-05	0.0034	3.5E-05
KNN	0.0130	2.4E-04	0.0119	1.8E-04	0.0059	5.2E-05	0.0032	1.4E-05
MR	no fit	no fit	no fit	no fit	0.0110	1.8E-04	0.0067	6.8E-05
MR Network	0.0129	2.3E-04	0.0127	2.2E-04	0.0115	1.8E-04	0.0069	7.2E-05
DNN	0.0139	2.7E-04	0.0114	1.8E-04	0.0055	4.9E-05	0.0041	2.1E-05
Turbine Decay								
SVR	0.0066	5.8E-05	0.0052	4.1E-05	0.0037	2.6E-05	0.0024	9.2E-06
DT	0.0073	8.5E-05	0.0050	4.9E-05	0.0046	3.6E-05	0.0028	1.5E-05
KNN	0.0067	7.0E-05	0.0051	4.3E-05	0.0052	4.8E-05	0.0033	2.2E-05
MR	no fit	no fit	no fit	no fit	0.0064	5.5E-05	0.0027	1.5E-05
MR Network	0.0067	6.0E-05	0.0068	6.3E-05	0.0065	5.7E-05	0.0063	5.6E-05
DNN	0.0067	6.7E-05	0.0045	3.4E-05	0.0035	2.3E-05	0.0034	1.4E-05