Maintenance Data Augmentation, using Markov Chain Monte Carlo Simulation

(Hamiltonian MCMC using NUTS)

Muhammad Ammar Roohani

Maintenance Engineering, master's level (120 credits)
2024

Luleå University of Technology
Department of Civil, Environmental and Natural Resources Engineering
[This page intentionally left blank]
Acknowledgments

In the name of Almighty, whose countless blessings and guidance has always been the beacon of light in pursuit of my endeavors throughout my life. I am extremely grateful to all the people whose crucial help and direction have made writing my thesis possible. It has been a truly amazing experience.

First and foremost, I would want to express my deep gratitude to Professor Uday Kumar, Head of the Department, for his kind support, allowing me to take a research project in the department. I would ever be thankful to my thesis supervisor Jaya Kumari, for her constant support, mentorship, and scholarly direction during this project. Her advice and support have been crucial in helping me develop this thesis and widen my horizons intellectually.

I would also like to thank Professor Ramin Karim, for his innovative and thought provoking insights on the subject of maintenance that intrigued me to take my thesis in the subject of maintenance analytics. Its been a pleasure and lovely experience studying at Lulea University of Technology and I would like to appreciate LTU for providing the favorable academic setting and resources necessary for conducting this research. The Department of Civil, Environmental and Natural Resources Engineering deserves special recognition for their unwavering support. In this regard, Johan Odelius, has been very kind and supportive in explaining the process and in helping the administrative matters.

I owe my family a debt of appreciation for their unwavering love, understanding, and support during my academic endeavor. My pillar of strength has been their support.
Contents
1. Introduction.......................................................................................................................... 3
   1.1. Research Problem ........................................................................................................... 5
   1.2. Research Objectives ...................................................................................................... 5
   1.3. Research Questions ......................................................................................................... 6
   1.4. Scope and Limitations ................................................................................................... 6
       1.4.1. Scope .......................................................................................................................... 6
       1.4.2. Limitations ................................................................................................................ 6
2. Theoretical Background and Concepts .............................................................................. 7
   2.1. Maintenance: a brief description .................................................................................... 7
   2.2. Types of Maintenance .................................................................................................... 8
       2.2.1. Planned Maintenance .............................................................................................. 8
       2.2.2. Corrective Maintenance .......................................................................................... 9
       2.2.3. Preventive Maintenance .......................................................................................... 9
       2.2.4. Condition-based Maintenance ................................................................................. 10
       2.2.5. Condition-based maintenance VS PM ................................................................. 11
   2.3. Maintenance Data Types .............................................................................................. 12
   2.4. Maintenance Data & Maintenance Analytics ............................................................... 13
       2.5.1. Feature engineering ................................................................................................. 15
       2.5.2. Method of Transfer Learning .................................................................................... 17
       2.5.3. Data Augmentation Techniques .............................................................................. 19
   2.6. Data Augmentation techniques on Graphs for table type classification ................... 20
   2.7. Gaussian Mixture Models (GMM) for Data Augmentation on CSV Files ................... 21
3. Research Methodology ....................................................................................................... 23
   3.1. Monte Carlo (MC) ......................................................................................................... 24
       3.1.1. Inversion Sampling ................................................................................................. 26
       3.1.2. The Fundamental Theorem and Rejection Sampling ............................................. 27
   3.2. Markov Chain Monte Carlo (MCMC) ........................................................................... 30
       3.2.1. Hamiltonion Monte Carlo Method ........................................................................... 31
       3.2.2. No-U-Turn Sampler (NUTS) Sampler .................................................................... 32
       3.2.3. PYMC Bayesian Inference Library ......................................................................... 33
       3.2.4. Some Comparison & Discussion about different MC Algorithms ......................... 33
   3.3. Dataset Details and Data Pre-Processing .................................................................... 34
   3.4. Time Between Failures Model ...................................................................................... 35
   3.5. Python Implementation ............................................................................................... 37
4. Results & Discussion ......................................................................................................... 39
5. Future Work ........................................................................................................43

Python Code .........................................................................................................47
Abstract

Reliable and efficient utilization and operation of any engineering asset require carefully designed maintenance planning and maintenance related data in the form of failure times, repair times, Mean Time between Failure (MTBF) and conditioning data etc. play a pivotal role in maintenance decision support. With the advancement in data analytics sciences and industrial artificial intelligence, maintenance related data is being used for maintenance prognostics modeling to predict future maintenance requirements that form the basis of maintenance design and planning in any maintenance-conscious industry like railways. The lack of such available data creates a no. of different types of problems in data driven prognostics modelling. There have been a few methods, the researchers have employed to counter the problems due to lack of available data. The proposed methodology involves data augmentation technique using Markov Chain Monte Carlo (MCMC) Simulation to enhance maintenance data to be used in maintenance prognostics modeling that can serve as basis for better maintenance decision support and planning.

1. Introduction

Each passing moment, the evolution and striding advancement in science & technology, heralds new marvels and every new achievement triggers the very curious nature of mankind to think and dig deeper to mark new milestones in its pursuit. From conceiving a seemingly novel idea of creating a wheel to inventing bullet trains, human mankind has come a long way, making its unbelievable mark in almost every sphere of science and technology. In today’s world, the striding advancement in science and development of sophisticated machines, has given confidence and belief in further possibilities. This pursuit has resulted in sophisticated and advanced realization of man-made systems in almost every sector of human society. With the inventions of advanced machines, there are also a swarm of unforeseen challenges when it comes to maintain such assets, so that, they keep on producing the desired outcomes efficiently and at a lower minimum cost. Because the performance of any technical asset does not only depend on its design, intensity of usage, and operating conditions etc., but the phenomenon of constant degradation affects its performance and other associated factors. Therefore, effective maintenance programs must be in place to ensure its smooth functioning. This led to a staunch necessity of ever developing and evolving science of maintenance regimes, that can be subtle in terms of technical effectiveness, efficiency, reliability and cost-effective to maintain the advanced and expensive technical assets.

Historically, only run-to-failure maintenance was in place where machines, as the name implies, where run until degradation had caused total termination of machines’ ability to perform the desired function and then maintenance personnel used to repair the machines to restore them in their working state. Some authors have stated this type of maintenance as corrective maintenance. Corrective maintenance (CM) is, thus, only carried out after a fault or malfunction or breakdown in a machine occurs (Wen et al., 2022a). Before world war II,
corrective maintenance was perceived as unavoidable cost and maintenance aspects were not even considered during design phase, let alone the consideration of business aspects. But sudden and uncertain breakdowns cause unscheduled downtimes, production delays, work overload, cost appreciation and many other associated problems with machines breakdowns. In comparison, Preventive maintenance (PM) is carried out at pre-planned intervals, even if the machines or components are in running condition, in order to avoid the uncertain failure due to degradation. Preventive maintenance is costly, as it does not take Useful Remaining Life (URL) of the equipment into consideration. Due to the increasing requirement of reliability, availability, maintainability, and safety of systems, preventive maintenance is becoming less effective and obsolete (Wen et al., 2022a). Up until recently, preventive maintenance has been the dominant maintenance policy in industries (Prajapati et al., 2012).

With advancement in measurement technologies, sensors, IoTs and industrial artificial intelligence, condition-based maintenance, also referred as predictive maintenance, is being deployed across the industries. It utilizes predictive tools to determine when maintenance actions are necessary and has drastically been adopted by industry as it helps in reducing maintenance cost, unexpected downtime, and while extending the life span of equipment (Wen et al., 2022a). In condition-based maintenance, a machine or a component is monitored or inspected continuously by employing non-intrusive testing techniques, such as thermodynamics, acoustics, vibration analysis, infrared analysis, etc. As degradation is monitored continuously, it gives asset’s health degradation trends in real time. Consequently, maintenance professionals can plan maintenance activities, resource optimization etc. The core procedures for implementing predictive maintenance include data collection, fault detection and diagnostics, and prognostics, which are later used to guide maintenance decisions (Wen et al., 2022). The identification of operational faults in any machine along with finding the root cause and effect of the fault is called “Diagnostics”. Whereas, Prognostics, in general, measures the extent of deviation and degradation of any machine or system from the normal operating behavior to predicts the Remaining Useful Life and future performance. (Wen et al., 2022a) It involves analytical computations of historical or real-time data streamed from applications, sensors, devices, etc.

Prognostics Modeling in Maintenance can be classified into two broader categories i.e.

- Physics-driven Prognostic Models
- Data-Driven Prognostics Models

The physical failure mechanisms are analyzed through physics-driven models and the degradation phenomenon is represented in the form of complex mathematical equations. This kind of approach is not only complex but also time consuming and sometimes impossible to adopt due to complexity of degradation phenomenon. Data-Driven Prognostics Models utilize the data that it receives from condition monitoring, failure data, reliability data etc. However, data driven models are easily adopted because these models can analyze anomalies and patterns in the data by mining the data and then prognostics model can also predict the future state of any system. These models basically utilize probabilistic research and open a world of possibilities in no. of ways, data can be used.

Most of the time, the conditioning data is available in abundance but there are cases where data related to failure times is not easily available. In such cases, where there is not enough data, prognostics model performance is affected in no. of ways. It has been discussed in another part
of the report in much detail. This research is related to the same domain of problems and tries to find the solution about lack of maintenance data or failure data that can be used in prognostics modeling to predict the future failure times, so that effective maintenance decisions can be made for efficient utilization of technical asset.

A limited amount of work has been done in data augmentation techniques by different researchers in the past, using different approaches and for different nature of data i.e. images, graphs, tables etc. However, no research has been found that may have utilized maintenance related data in the form of rows and columns like in excel files, to be augmented and used in prognostics modeling. The proposed methodology involves maintenance data augmentation using Markov Chain Monte Carlo (MCMC) Simulation, using maintenance data related to time to failure of braking system of a set of trains. The input to the model is time between failures (TBF) and total of 114 failure times are available for braking system failure in the dataset and it clearly shows that this is not enough data that can be utilized in any probabilistic modeling, since insufficient amount of data entails lots of other problems (discussed later) that affect the prognostics modeling. Therefore, MCMC simulation is being utilized to synthesize more data samples to be augmented to original dataset to make it sufficiently large in order to get better model performance that can be utilized in prognostics modeling.

1.1. Research Problem

The importance of sufficient amount of data is detrimental to create results, analyze, detect patterns, and foresee the future outcomes in any field of business, engineering, and research. When it come to maintenance of technical assets specifically, its importance and utilization is hinged on multiple dimensions i.e. type, nature, amount and mechanism through which data is being obtained. However, as science and engineering are witnessing a tremendous advancement, technical/ engineered systems are becoming more sophisticated and reliable that are more efficient and less prone to faults and failures. This is also because of improved maintenance regimes employing modern tools like advanced condition monitoring that is monitoring the health of any technical system round the clock in order to mitigate any risk of failure before it happens. In such situations, the availability of failure times data becomes very difficult. However, there is still the requirement of availability of such data for maintenance analytics and to predict the failure in future. There has been a lot of research work to make up for the less amount of data by utilizing different methods that are discussed further in this document and they are mostly focused on imagery and in some cases tabular or graph data. But there has not been enough work upon any suitable methodology to make up for less amount of data in the form of an excel or csv files i.e. in the form of columns and rows. Mostly, maintenance related data is available in numerical data form i.e. in rows and columns in an excel file. Depending upon our focus on maintenance data, a new method of data augmentation needs to be interrogated in order to analyze its suitability in maintenance analytics.

1.2. Research Objectives

The purpose of this thesis is to develop a methodology that is useful in enhancing the performance of prognostics models in maintenance.
The research objective is to develop a data augmentation technique for Maintenance Data in the form of rows and columns, using Markov Chain Monte Carlo Simulation, so that data-driven maintenance prognostic models performance can be improved using additional synthesized data.

1.3. Research Questions

Based on the literature survey of data augmentation techniques in Maintenance Analytics or Maintenance Data, the following research questions were considered:

1. Can Data Augmentation of Maintenance Data, using proposed method, enhance the quality of maintenance data to be deployed to test the performance of prediction/prognostic models?
2. What are different data augmentation techniques for different types of maintenance data?

1.4. Scope and Limitations

1.4.1. Scope
1. The methodology is developed to perform the data augmentation of numerical data only and does not consider associated reasons for failures.
2. The effects of failures leading to failures in braking system were not considered.

1.4.2. Limitations
2. The underlying science and mathematics behind the Markov Chain Monte Carlo Simulation has not been discussed in detail because the primary focus is not the evaluation of probabilistic mathematics but the results only.
3. No prediction algorithm was developed in this study, rather, an already existing algorithm was used to compare the results.
2. Theoretical Background and Concepts

2.1. Maintenance: a brief description

The field of maintenance is as old as the history of human’s technical inventions. Since then, it has been evolving and taken a form of a dedicated science.

According to International Standard IEC 60300-3-14, maintenance is defined as:

“*The combination of all technical and administrative actions, including supervision actions, intended to retain an item in, or restore it to, a state in which it can perform a required function*.“ (Dependability Management-Part 3-14: Application Guide-Maintenance and Maintenance Support, 2004)

The set of actions taken in the light of above-mentioned definition are called maintenance activities and their ultimate goal is to either retain an item or restore it to its functioning state while taking care of time and cost as well. These actions encompass all the dimensions of administration as well as technical initiatives that are designed and executed according to a well-thought-out policy, scope and strategy in order to achieve ‘maintenance objectives’. The maintenance objectives may vary from industry to industry but core objectives of any organization because of effective maintenance program remain universal. Such maintenance objectives have been summarized in Figure 2. (Mohammed Ben-daya et al., n.d.).

The maintenance actions performed to get maintenance objectives vary in terms of their complexity, design and nature of execution and depend upon the type of equipment being maintained. They can be performed at any stage of the decomposition of engineering object. For example, maintenance can be carried out at Plant level i.e. periodic shutdown of power plants to access the health of the whole system made up of engineering objects, to repair or replace the damaged/ degraded equipment. Similarly, maintenance actions can be performed on Intermediate level i.e. subassemblies or modules (replacement of a motor or an electronics card from a machine). The maintenance actions performed on Component level involve basic maintenance actions focused on a component i.e., replacement of a degraded belt, or replacement of a bearing in a motor which is considered as a component.
Therefore, depending upon the scale, complexity and requirement, maintenance has been
categorized into different classifications. The following figure explains the broader
maintenance types that have been developed over the course of industrial evolution. In very
early times, such a concept of different types of maintenance did not exist and only run-to-
failure maintenance was in place when it was only required to repair or replace a component
or machine if it would go out of operation (Mohammed Ben-daya et al., n.d.; Wen et al., 2022b).

![Figure 2 Major Maintenance Objectives](Mohammed Ben-daya et al., n.d.)

2.2. Types of Maintenance

Maintenance can be broadly classified into two groups i.e. Planned & Unplanned Maintenance.
Planned Maintenance includes Preventive Maintenance (PM), Corrective Maintenance (CM),
Predictive Maintenance (CBM) and Improvement Maintenance (IM) and here is a brief account
of main types only.

2.2.1. Planned Maintenance

As the name explains itself, PM actions prevent the occurrence of failure before machine
actually breaks down. PM actions are carried out according to predefined set of criteria in terms
of time, usage or condition and are intended to reduce the probability of failure or the functional degradation of an item (Mohammed Ben-daya et al., n.d.).

2.2.2. Corrective Maintenance

Corrective maintenance restores the functions of an item after failure has occurred or performance fails to meet stated limits (Dependability Management-Part 3-14: Application Guide-Maintenance and Maintenance Support, 2004). The maintenance actions carried out after fault recognition to restore a failed item into a functioning state to perform its normal function. Sometimes there is an option either to replace a failed item with a new or refurbished or repair it. The choice of doing either one of them depends on many things like criticality of the equipment, resources and budget, technical expertise required to repair and time etc. For example, it is easy to replace a bulb in a lamp and cheap too, instead of getting it repaired and install it. Similarly, it’s not advisable to fix a faulty bearing of car wheel. Instead, it is replaced with a new one. It is comparatively cheap and quick.

Corrective Maintenance can be further classified into two categories (Mohammed Ben-daya et al., n.d.):

- **Immediate CM**: To be carried out immediately to restore an item to a functioning state to avoid expensive loss of production, environmental damage, safety breach and further damage of the equipment.

- **Deferred CM**: Deferred CM is corrective maintenance which is not immediately carried out after fault detection but is delayed in accordance with given maintenance rules (EN 13306, 2001). In such a case, it is affordable in terms of safety, loss in production etc.

![Figure 3 Maintenance Classification](Mohammed Ben-daya et al., n.d.)

2.2.3. Preventive Maintenance
Preventive maintenance may be carried out at regular intervals or according to prescribed criteria to reduce the probability of failure or degradation in order to retain the functioning of an item or to detect a hidden fault (Dependability Management Part 3-14: Application Guide-Maintenance and Maintenance Support, 2004). Preventive maintenance actions are performed on an operational equipment and after taking it out of the service to enhance the equipment lifetime and reliability (Prajapati et al., 2012). It is performed to prevent failures because failures can be costly in terms of financial, environmental and safety concerns. Sometimes PM actions help detect the potential failure in advance and it sets the preparatory actions to perform maintenance as soon as failure occurs. This may save time and cost while allowing the maintenance teams for better resources utilization.

Preventive maintenance actions are usually grouped into three categories i.e. Predetermined, Condition-based and opportunistic, as described below.

<table>
<thead>
<tr>
<th>Pre-determined</th>
<th>Condition-Based</th>
<th>Opportunistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Time Based</td>
<td>• Degradation Assessment through condition monitoring</td>
<td>• Failure or idle machines providing an opportunity to maintain PM actions on non-failed components</td>
</tr>
<tr>
<td>• Usage Based</td>
<td>• Change of parameters</td>
<td></td>
</tr>
</tbody>
</table>

### 2.2.4. Condition-based Maintenance

Condition-based maintenance is a part of proactive maintenance and yet an advanced type of maintenance where a condition monitoring apparatus monitors the condition of the equipment in terms of degradation and performance, either continuously or at regular discrete time intervals. The objective is to measure a measurable parameter that correlates to degradation of the equipment and the same parameter when measured over time, is compared, and analyzed to predetermined check if the equipment has gone into degradation and to what extent. The measurable parameter does not mean that an intrusion can be made to maintain the asset but it’s the preventive action that is allowed to carry out after measuring the parameter that indicates any degradation. This field of maintenance has been evolving tremendously and advanced measurement technologies as well as advanced analysis into the health of any technical system through measurable parameters helps you gain deep insights about the technical condition of any equipment.
For example, in railways, different critical components in train vehicles and train track degradation are monitored through condition monitoring tools to see whether they are being degraded due to external or internal factors. Such an approach of condition monitoring provides substantial time and opportunity to plan the maintenance in advance. The figure explains the concept of condition-based maintenance where \( t_1, t_2, t_3, \ldots \) are the time intervals of monitoring the equipment after equal intervals to check the degradation. As soon as the degradation level falls below the red line for Alarm threshold, the time interval between \( t_a \) and \( t_o \) provides an opportunity to carry out preventive maintenance to avoid complete failure and carry out the corrective maintenance to restore the equipment for functioning again.

Conditioned-based maintenance can be of two types (Mohammed Ben-daya et al., n.d.):

- **Function Tests**
- **Condition Monitoring & Inspection**

Function tests aim to check the condition of the equipment at a particular moment to decide if the maintenance is required or not, however, the latter is to carry out interval checks and to take measurable parameter values. This data is usually in the form of frequencies and measured through sensors (Prajapati et al., 2012). The data can then be used in advanced machine learning models to predict the failure time in future or any other sort of analysis. (Predictive Maintenance)

### 2.2.5. Condition-based maintenance VS PM

Preventive Maintenance allows to replace the component in an equipment after a predetermined time or usage to retain the functioning of the equipment without risking the total failure, thus, discarding the remaining useful life of the component. While, condition monitoring allows to take control of deciding whether the equipment need PM or not and fully utilizes the remaining useful life of the equipment or a component. Hence, it reduces the maintenance costs, save time, and assists in efficient maintenance management.

Following is the comparative chart for broader types of maintenance and their applications, requirements, advantages, and disadvantages.
2.3. Maintenance Data Types

Data, in its different forms and nature, is utilized across the globe for churning out information and knowledge through processing of raw data. Seemingly never-ending and vastly expending industrial innovations and inventions resultanty churn out loads of data in different forms that gets processed and manifests itself into knowledge and wisdom. Data has, in fact, become the most sought-after and valuable commodity in these modern times. When it comes to maintenance data, it has various forms and types, depending upon the different scenarios. There are different types, classification and nature of data that mostly depend upon the type of equipment being monitored or maintained. Equipment performance & condition-based data provides valuable information about uptime, downtime, current state of degradation of different equipment components etc. while time-based data is any data that is categorized with respect to time i.e. scheduled maintenance, date and time of repairs etc. Failure Data of any equipment is mostly sought-after, and it comprises of the information related to date and time of failure, failure type and related information that can help identify the root cause of the failure. Maintenance data can also be sometimes structured or unstructured. The structured maintenance data has closes ended information while unstructured data can be in the form of descriptions and comments. For example, comments made by a maintenance personnel about the corrective maintenance action are descriptive and unstructured. To carry out a qualitative or quantitative analysis, the descriptive maintenance data can be structured using natural language processing technique, also called Text Tagging or Annotations.
The following fig. Shows different classifications of data:

![Figure 6 Maintenance Data Classification](image)

### 2.4. Maintenance Data & Maintenance Analytics

Different types of data have different form of utilization that range from operations and maintenance management, planning, diagnostics, and prognostics modeling etc. Failure times data is used in reliability calculations & prognostics modeling to predict the future failures and supplementary data may help in maintenance costs calculations, maintenance planning and design as well as maintenance decision support. Maintenance Data is also used to get unseen information and knowledge by recognizing special patterns of data which leads to knowledge discovery. Therefore, Maintenance data is vital in maintenance decision support. The knowledge discovery, which is an essentially a major aspect for maintenance decision support; is usually done by discovering special pattern of data, i.e., by clustering together data that share certain common properties (Wang, 1997). Knowledge discovery through analyzing maintenance data opens new frontiers of possibilities that assist in maintenance decision support effectively. The processing of maintenance data to generate useful information and knowledge has been termed as “Maintenance Analytics” by (Karim et al., 2016). MA focuses on the new knowledge discovery in maintenance. MA addresses the process of discovery, understanding, and communication of maintenance data from four time-related perspectives, i.e. 1) “Maintenance Descriptive Analytics (monitoring)” focuses to discover and describe what happened in the past; 2) “Maintenance Diagnostic Analytics” focuses to understand why something happened; 3) “Maintenance Predictive Analytics” focuses to estimate what will
happen in the future; and 4) “Maintenance Prescriptive analytics” which addresses what need to be done next. (Karim et al., 2016)

However, there comes some scenarios where the creation & availability of certain data becomes unfavorable at the cost of events that produce such data in the first place. An engineering asset, when efficiently maintained, does not fail frequently and in result, available failure times are not sufficient to use in data driven prognostics modeling. For a better maintenance decision support, prognostics modelling and maintenance management, only enough data related to a machine’s performance, failure and its health, can help carrying out well-crafted analytics that can now-cast or forecast any undesirable events or failures in particular. Such an analysis in Maintenance is termed as ‘Maintenance Analytics’ that is utilized in diagnosis, prognosis, and better health management of engineering assets; while the inherent phases of Maintenance Analytics are highly dependent on availability of vast amount data from various data sources(Karim et al., 2016).

2.5. Data Driven Models for Maintenance Analytics

The importance of sufficiently available large amount of data is because of its utilization I data driven models, instead of physics driven models. Data driven models have a no. of advantages over physics driven models due to data availability, simplicity and flexibility to play with the data in data driven models; whereas, the physics driven models employ complex mathematical equations that can be computationally challenging to solve in terms of speed and cost (Zhang et al., 2019). In order to solve utter complexities in data driven models and enhance efficiency, Prognostics Modeling are utilized. Predicting machine failures, performance, prognosis and optimization of engineered systems require analysis of data and Prognostics Modeling in all this activity takes a valuable importance.

Since data driven models performance in maintenance prognostics modeling can be significantly affected if there is no sufficient amount of data present. Data scarcity impacts the model performance negatively due to a no. of problems like overfitting, underfitting, limited model selection, poor generalization, limited feature space, Skewed models and limited data driven decisions making etc. When there are fewer examples for the model to learn from due to data scarcity, it is trained to fit the noise instead of underlying patterns and overfitting the
model with less concerned data results in poor model performance as chances of overfitting are greater with small datasets (Li et al., 2021).

In order to mitigate the effects of low amount of data as stated above, different techniques have been employed by the researchers and data scientists to take full advantage of available data. Methods like feature engineering, regularization techniques, cross validation, transfer learning and data augmentation etc. are few to name that are utilized to solve the problems related to low amount of data (Maharana et al., 2022). Few such methods have been discussed here.

2.5.1. Feature engineering

The process of creating new features or transforming existing ones in a dataset to improve the performance of machine learning models, is called Feature engineering. Raw data is selected, combined and transformed to create different but informative and predictive features in addition to the existing dataset, in order to obtain relationships and patterns in the data. The process using feature engineering helps in:

1. Improvement of model performance: By creating relevant features, we can provide the model with more meaningful information, enabling it to learn better and make more accurate predictions. (Khalid, 2014)
2. Capturing complex relationships: Feature engineering allows us to encode domain knowledge and capture complex relationships between variables that might not be apparent in the raw data. This can significantly enhance the model's ability to generalize and make accurate predictions. (Fan et al., 2019)

3. Dimensionality Reduction: Feature engineering helps in selecting or creating a subset of features that are most relevant to the problem at hand. This can reduce the dimensionality of the data and remove noise or redundant information, leading to improved model performance and faster training times. (Khalid, 2014)

Feature Engineering is an effective tool when amount of data is not sufficient and it can provide more data so that meaningful information can be extracted that depicts important characteristics of the problem. For example, in image classification tasks, features can be extracted from pre-trained deep learning models and use them as inputs to your own model. Instead of relying solely on individual features, improved features can be created by combining or aggregating existing ones. This can help in capturing higher-order relationships or summarizing information across different dimensions. Similarly, scaling or normalizing features to a common range can be crucial when dealing with limited data. It ensures that all features have similar ranges, preventing some features from dominating the learning process. In situations with limited data, selecting the most informative features becomes even more important. Techniques like correlation analysis, mutual information, or L1 regularization can help identify the most relevant features to include in the model.

Overall, feature engineering plays a vital role in making the most of available data, regardless of its quantity. It helps in extracting valuable information, reducing noise, capturing complex relationships, and improving the performance of machine learning models.

2.5.1.1. Limitations of feature engineering application w.r.t. data types

Feature engineering techniques are suitable for various types of data, including numerical, categorical, and textual data. Here's a breakdown of the suitability of feature engineering techniques for different data types:

1. Numerical Data: Feature engineering techniques can be used to create new features or transform existing numerical features. For example, you can apply mathematical
operations (e.g., logarithm, square root) to scale or normalize the data, create interaction terms, or derive statistical features such as mean, standard deviation, or percentiles.

2. Categorical Data: Categorical data can be encoded using techniques like one-hot encoding, label encoding, or target encoding. These techniques transform categorical variables into numerical representations that can be effectively used by Prognostics Modeling. Additionally, feature engineering can involve creating new features based on categorical variables, such as aggregating or counting occurrences of certain categories.

3. Textual Data: Feature engineering techniques for textual data involve transforming raw text into numerical representations. This can include techniques like bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), word embeddings (e.g., Word2Vec, GloVe), or more advanced methods like BERT (Bidirectional Encoder Representations from Transformers). These techniques extract meaningful information from text data that can be used as features for machine learning models.

2.5.1.2. Data Augmentation Vs Feature Engineering

Regarding data augmentation, feature engineering has limitations in its direct applicability. Data augmentation typically focuses on generating additional synthetic samples by applying transformations to existing data. Feature engineering, on the other hand, involves creating or transforming features rather than generating new samples. While feature engineering can indirectly contribute to data augmentation by providing more informative and diverse features, **it does not directly address the need for increasing the quantity of data.**

Data augmentation techniques such as rotation, scaling, flipping, or adding noise to images, or perturbing text with synonyms or similar variations, are typically not considered traditional feature engineering techniques. Instead, they are used to increase the diversity and variability of the available data, helping models generalize better and reduce overfitting.

In summary, while feature engineering techniques are suitable for various types of data, their direct relationship with data augmentation is limited. Feature engineering focuses on creating or transforming features, while data augmentation techniques aim to generate additional synthetic samples to enhance the diversity and quantity of the data. Both approaches, however, can work together to improve the performance of machine learning models.

2.5.2 Method of Transfer Learning

Method of Transfer Learning helps avoiding dependence on large datasets, in which a model is trained on a different dataset and then retrained on a different dataset, related to the same type of domain, by adjusting the weights on the trained model. Transfer learning aims at improving the performance of target learners on target domains by transferring the knowledge contained in different but related source domains (Zhuang et al., 2021). In machine learning models, where there has been a tremendous evolution in terms of methodologies and applications, however, their application continues to pose challenges when it comes to their application on real world data because most of these models run on supervised training instances on mass labeled datasets.
Therefore, obtaining large amount of labelled data is cumbersome in terms of time constraints, expenditure as well as sometimes it is unrealistic. However, semi-supervised learning models approach solve the need of abundant data and partly makes use of labelled data and then utilizes the large amount of unlabeled data to improve model performance of the learning. But in some cases, getting unlabeled data is also difficult to collect and thus it makes traditional learning models inaccurate.

The method of transfer learning, in simplest terms, implies that if a person can drive a car, then his or her driving skills can be transferred to learn driving bigger vehicles. The same way, if a musician can play a guitar, then he can learn to play the piano, which is another music instrument and both of these skills falls in the same domain.

2.5.2.1 Transfer Learning Categorization

The approach to transfer learning can broadly be categorized according to the problem and solution, which further can be classified into Label-setting based and space setting-based categorizations. These two divisions in approaches are further divided into multiple groups of categorizations as shown in the fig.2. For example, if the information regarding label comes from source domain, then the method of transfer learning can be named as Transductive. However, the label information is only coming from target domain, it is called inductive. But if the information is neither coming from source nor from target domains, the categorization of transfer learning is called unsupervised transfer learning (Zhuang et al., 2021).
The transfer learning approaches can be categorized into four groups: instance based, feature-based, parameter-based, and relational based approaches. Instance-based transfer learning approaches are mainly based on the instance weighting strategy. Feature-based approaches transform the original features to create a new feature representation; they can be further divided into two subcategories, that is, asymmetric and symmetric feature-based transfer learning. Asymmetric approaches transform the source features to match the target ones. In contrast, symmetric approaches attempt to find a common latent feature space and then transform both the source and the target features into a new feature representation. The parameter-based transfer learning approaches transfer the knowledge at the model/parameter level. Relational-based transfer learning approaches mainly focus on the problems in relational domains. Such approaches transfer the logical relationship or rules learned in the source domain to the target domain.

However, this method is redundant when the model has to be trained on a dataset of different type and incurs negative transfer learning as well. (Zhuang et al., 2021)

2.5.3. Data Augmentation Techniques

One of the effective methods to get more data of similar kind is Data Augmentation in which synthetic data is produced to be later incorporated to the original dataset, thus enhancing its size considerably. The contemporary field of application for data augmentation techniques is visual imagery. And contemporary methods are cropping, flipping, rotating and merging etc. to create more samples of images. However, Random Erasing technique to create more sample of image data has been used as well in which pixels of the images are erased randomly and different settings of erasing is applied to create a dataset to train Convolutional Neural Networks (Zhong et al., 2017). Similarly, other several data augmentation techniques have also been used for image processing that have contributed a lot in medical applications. However,
data augmentation techniques for statistical data in the form of columns and rows (CSV & Excel files) has not been explored much.

(Snow, n.d.) developed a novel Tabular data augmentation (TDA) Technique which is a process of modular feature engineering and observation engineering for tabled data, emphasizing upon the order of augmentation to achieve the best predicted outcome for a fixed information set. But the technique aims to improve the quality of data by contemporary methods of mapping, classification and transformation which cannot be considered for the kind of maintenance data which is already qualitative but not quantitative.

2.6. Data Augmentation techniques on Graphs for table type classification

Tables, in the form of rows and columns, in any document smartly organize information that helps their readers to grasp the presented information in a quick and well understood manner. Tables provide a wide range of angle to understanding of information like comparisons of results, summary of information, classification and categorization of information with associated knowledge in the form of nos., codes, symbols, and descriptive information etc. However, the importance of tables in scientific papers is paramount since it helps researchers organize information and compare results efficiently, but, tables are formulated in different layouts and types that there is no uniformity among them and sometimes they are irregular and contain abbreviations and symbols. Some significant work has been done in the past to design methodologies to extract tables from scientific journals and documents and then further extract information, without need to go through the whole document. It is because most of the times, researchers are concerned about knowing and comparing the research results and then making use of it further.

The first steps to address this issue is to detect and then define the structure of tables from a scientific paper or document. This has previously been done using computer vision and natural language processing techniques. Then, few advanced mechanisms have also been developed, combining vision, semantic and relations for layout analysis and table detection(Zhang Peng and Li, 2021) and applying a soft pyramid mask learning mechanism in both the local and global feature maps for complicated table structure recognition(Qiao Liang and Li, 2021).
Using a different approach, (del Bimbo et al., 2022) developed a model using Graphical Neural Networks (GNNs) to classify the scientific tables according to their types because GNNs are considered widely in document analysis and table understanding and then using data augmentation technique in applied data augmentation technique on graphs for table type classification in which tables from scientific papers were extracted and classified through Graphical Neural Networks (GNN). After classification, random inversion of rows and columns as well as random deletion of edges and nodes were performed to create new graphs. However, this type of data augmentation is not suitable for maintenance data in statistics and numerical values. This method also has its limitations when working with types of tables.

In another sphere of similar kind, (T. Tran et al., n.d.) used Bayesian formulation for image classifications and data augmentation of images only, along with Monte Carlo Expectation-Maximization Algorithm for learning. Since there are many other such example available in the literature where data augmentation has been performed mostly on images, it has not been discussed here because our research focus is only on data augmentation of maintenance data in the forms of rows and columns i.e. Numerical data.

2.7. Gaussian Mixture Models (GMM) for Data Augmentation on CSV Files

(Arora et al., 2021) used Gaussian Mixture Models (GMM) to generate synthetic data with CSV files. It is a probabilistic model which learns the distribution of the data, and then samples out the artificial examples based on this distribution. It was a promising effort, however, the techniques has not yet been validated for CSV files of different kind of datasets (Maintenance related Data). Further, maintenance data which depends on no. of factors, does not necessarily portray a normal gaussian distribution, therefore, rendering this technique not useful in this case.
This thesis aims to experiment with a methodology that can add more data points (data augmentation) to already available maintenance data, regarding time between failures of Braking System of a trainset through data augmentation technique using Markov Chain Monte Carlo simulation and comparing the results by testing original data as well as augmented data, through already existing failure prediction algorithms. This research is mainly focused on maintenance data and sheds the light on different types of maintenance data, its utilization, and challenges due to insufficient amount of data for maintenance analytics.

This Data Augmentation Technique utilizes Markov Chain Monte Carlo Simulation (NUTS Algorithm) to draw random samples from a distribution coming from maintenance dataset, containing Time Between Failures in Days. MCMC simulation shall be utilized to generate additional samples from the posterior distribution of the model parameters, which can then be used to create a larger dataset for training or testing purposes. MCMC method is also useful to implement if it is difficult to draw samples from an unknown distribution. The nature and variation in maintenance related data resembles the same attributes of randomness and uncertain distributions sometimes. A comparison of results shall be achieved by evaluating different data distribution parameters. It is expected to get improved results from such qualitative comparison.
3. Research Methodology

The research methodology is shown in the following fig. It consists of broadly two parts. The first part related to research study that includes the literature survey, identification of research questions, scope & limitations, and theoretical background related to different types of maintenance and maintenance data. Second part relates to practical implementation of proposed model i.e., MCMC simulation for maintenance data augmentation. It starts with the pre-processing of the data, present in excel file, and calculation of time between failures (TBF) which were used in MCMC simulation on Python.

Results and Analysis includes results from simulation, comparison of results obtained after running prediction model for original and augmented data and finally the conclusion and discussion on model performance.

![Figure 15 Research Methodology](image-url)
3.1. Monte Carlo (MC)

Markov Chain (MC) is a mathematical technique to estimate the possible outcomes of uncertain events which are otherwise difficult to predict due to introduction of independent variables. Monte Carlo methods may be thought of as a collection of computational techniques for the (usually approximate) solution of mathematical problems, which make fundamental use of random samples and two classes of statistical problems are most commonly addressed within this framework: integration and optimization (Johansen, n.d.). The name "Monte Carlo" refers to the city in Monaco, known for its casinos and gambling.

The term "Monte Carlo" is essentially a synonym for randomness, and Monte Carlo simulations are simulations that purposefully evolve in a random manner. The fundamental tenet of a Monte Carlo simulation is that, if we let the simulation develop at random, we can get a representative sample of a huge population of possible outcomes. A population can be defined as a set of examples and a sample is a proper subset of a population. In simple terms, a random sample tends to exhibit the same properties as a population from which it is drawn.

**Theoretical Example**

While there is some controversy about the nature of the first Monte Carlo computations ever performed (with some authors claiming that they date back to the time of ancient Babylon), it is generally agreed that the earliest modern Monte Carlo experiments took place in the later decades of the nineteenth century and were concerned with calculating π (Johansen, n.d.-c).

In one Monte Carlo experiment, rain is assumed to fall evenly at random over a square region of space, with a circle
enclosed by the square. The location of each raindrop may be viewed as a realization of a uniformly distributed random variable. It follows intuitively from no formal probability theory that the likelihood of a uniform raindrop falling in any area of the square must be proportionate to that area's area and unaffected by its position. As a result, their areas can be used to describe the probability, \( p \), that a raindrop is inside the inscribed circle. If the square has sides of length \( 2r \), the circle must be of radius \( r \) and:

**Equation 1. Calculating probability for raindrops inside the inscribed circle (example)**

\[
p = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4}
\]

This might not appear extremely intriguing. Nonetheless, \( \pi \) can be roughly approximated by using its estimators after being stated as a function of this probability. Analytical thinking is not possible because knowing \( \pi \) is necessary to calculate the probability. It is intuitively possible to calculate this probability by counting the proportion of raindrops that fall inside the circle. For example, if \( n \) raindrops are observed and \( m \) of them fall inside the circle, one may estimate probability using the formula:

\[
\hat{p} = \frac{m}{n}
\]

Assuming that there were 500 raindrops in total, uniformly distributed over the square region as shown in the fig. The computer simulation shows that 383 raindrops fall in the circle that is inscribed in a square. This implies that \( \hat{p} = 383/500 \) and comparing this with the earlier expression in terms of \( p \) and \( \pi \), the value of \( \pi = \frac{383}{125} = 3.06 \). Although this is not an accurate value, but it is just an estimation.

Confidence in the above estimation depends on two factors:

- Size of the sample
- Variance of the sample

As the variance grows, we need larger samples to have the same degree of confidence in our estimation, Lesser the variance, greater the confidence in estimation. In the above example, if the simulation is performed a no. of times, and taken average of the outcome everytime, the estimated value of \( \pi \) would get closer to it real value.

The above phenomenon is based upon the **Law of Large Nos**. According to the law, the average of the results obtained from a large no. of trials should be close to the expected value and tends to become closer to the expected value as more trials are performed(Pred, n.d.).

This means that the probability of any independent outcome tends to get to its true value when samples are taken repeatedly in completely random fashion and when taken the mean, it tends to approach its real value. This also implies that variance decreases close to zero. The more samples you take, the more likely you get to the mean. It is important to be able to sample from the distribution of interest in order to use simple Monte Carlo. This criteria is not always readily met: Monte Carlo methods are typically employed to deal with intricate distributions that do not permit tractable analytic solutions, and obtaining samples from such distributions can be quite challenging. Following the introduction of two generally applicable strategies for getting samples from known distributions, a method for estimating expectations under a distribution
using data from a different distribution is presented. That is, the samples obtained using the following two ways for sampling from a distribution of interest can be employed in the simple Monte Carlo method described above. (Johansen, n.d.-c)

3.1.1. Inversion Sampling

(Johansen, n.d.-c) has explained Inversion sampling in an easily understood manner. The Cumulative Density Function (CDF), \( F(x) \) can be defined as under for a Probability Density Function (PDF), \( f \):

Equation 2. CDF for a given PDF

\[
F(x) = \int_{-\infty}^{x} f(y) \, dy
\]

If \( X \) is distributed according to \( f \), then \( F(x) = P(X \leq x) \). The generalized inverse of a distribution function, \( F \), may be defined as:

Equation 3. Inverse of a distribution function

\[
F^{-1}(p) = \inf \{ x : F(x) \geq p \}
\]

That is, it is the smallest value of \( x \) for which \( F(x) \) is greater than or equal to \( p \). This varies from the inverse only if \( F \) is discontinuous (which occurs anytime the related random variable has a finite probability of adopting a particular value). By setting \( X = F^{-1}(U) \), inversion sampling converts a random variable, \( U \), with a uniform distribution on the interval \([0, 1]\) (i.e., a random variable with density 1 over \([0, 1]\) and 0 everywhere). If \( F \) is the generalized inverse of the distribution function of interest, \( X \) has the following distribution:

Equation 4. Distribution of variable

\[
P(X \leq x) = P(F^{-1}(U) \leq x)
\]

Figure 18 Graphical Example of Law of Large Numbers (Johansen, n.d.-c)
\[ P(U) \leq F(x) = F(x) \]

The procedure is depicted in Figure. The three realizations of \( U \) are represented by the hollow circles on the vertical axis, while the three realizations of \( X \) are represented by the filled circles on the horizontal axis. If the inverse of the distribution function is known, this method can be used to sample realizations of any one-dimensional real-valued random variable (Johansen, n.d.-c).

### 3.1.2. The Fundamental Theorem and Rejection Sampling

A more general approach is motivated by the fact that sampling a set of random variables based on a particular density is identical to sampling uniformly throughout the area under the density graph while ignoring the additional dimension. Saying that a real-valued random variable, \( X \), has density \( f \) indicates that the probability that \( X \) is between \( x \) and \( x + dx \) is \( f(x)dx \) for infinitesimal \( dx \). If we divide the region between \( f(x) \) and the x-axis into infinitesimal squares of uniform area and then choose one at random, we will get a sample for \( X \) with the same distribution (the no. of boxes in the strip of interest is, of course, proportional to the no. of boxes in the strip of interest) as illustrated in fig. (Johansen, n.d.-c).

More formally, sampling uniformly from the region \( \{ (x, u) : 0 \leq u \leq f(x) \} \) and retaining only \( x \) is equivalent to sampling \( x \) according to \( f(x) \). This is sometimes known as the fundamental theorem of sampling. Of course, one cannot usually sample uniformly from this region by any direct means – if anything, it is more difficult than sampling from the distribution of interest. Rejection sampling provides samples uniformly under the graph of the density of interest by sampling from a larger area and rejecting those samples which fall outside the region of interest.
Formally, given a density, $g$, from which it is possible to obtain samples and some known constant, $M \geq \sup_x \frac{f(x)}{g(x)}$, one can consider the following algorithm to obtain a sample from $f$:

1. Sample $X$ from $g$.
2. Sample $U$ from $\mathcal{U}(0, M)$;
3. If $U > f(X)/g(X)$ reject $X$ and go to 1.
4. Accept $X$ as a sample from $f$

Figure illustrates the principle: the filled circles indicate accepted samples (which are distributed uniformly beneath the bimodal density) and the hollow circles were rejected (samples were generated uniformly in the area under the rescaled unimodal density).
Notice that, on average, one sample in $M$ is accepted:

\[
P(X \text{ Accepted}) = P \left( U \leq \frac{f(X)}{g(X)} \right) = \frac{1}{M} \cdot E \left( \frac{f(X)}{g(X)} \right) = \frac{1}{M}
\]

In order for this to be an efficient strategy, it must be possible to sample from a distribution for which a small value of the constant $M$ can be found. To confirm that the distribution of the accepted samples is, indeed, $f$, it is sufficient to take into account the probability that a sample lies in an infinitesimal neighborhood, $dx$, of a point $x$ given that it is accepted.

\[
P \left( X \in dx \mid U \leq \frac{f(X)}{g(X)} \right) = \frac{P \left( X \in dx, U \leq \frac{f(X)}{g(X)} \right)}{P \left( U \leq \frac{f(X)}{g(X)} \right)} = P(X \in dx) \cdot P \left( U \leq \frac{f(X)}{g(X)} \mid X \in dx \right) / (1/M) = g(x)dx \cdot \frac{1}{M}f(x) / g(x) \cdot M = f(x)dx
\]

A approach for lowering the computational cost of samples produced via rejection sampling is the envelope or squeeze method. It is possible to automatically accept samples for which $U$ is less than the lower bound and reject samples for which it exceeds the upper bound if there are
inexpensive upper and lower constraints for the ratio \( f(x)/g(x) \). As a result, only when the sampled value of \( U \) falls inside the range between these two limitations is it necessary to assess the ratio itself, which could take some time. If the boundaries are properly selected, this can result in significant computational savings. (Johansen, n.d.-c)

### 3.2. Markov Chain Monte Carlo (MCMC)

In general, a discrete-time Markov chain is a group of random variables with a temporal ordering and the property that, conditional upon the present, the future is independent of the past. (Johansen, n.d.-a) This idea, which can be seen as a form of the Markov property, is formalized by the statement that a sequence of random variables, \( X_1, X_2, \ldots \) forms a Markov chain if and only if the joint pdf of the first \( n \) elements of the sequence can be broken down in the following way for any value of \( n \):

\[
p(x_1, \ldots, x_n) = p(x_1)p(x_2|x_1)\ldots\ldots p(x_n|x_{n-1})
\]

Even while a class of continuous-time stochastic processes can be developed with a similar lack-of-memory trait, these are rarely applied in an MCMC setting.

The fundamental idea behind MCMC is that if a Markov chain can be built such that a series of draws from the chain have statistical properties that are somewhat similar to a collection of draws from an interest distribution, it is also possible to estimate expectations with respect to that distribution by using the standard Monte Carlo estimator but using the dependent collection of random variables obtained by simulating a Markov chain instead of an independent collection of random variables. (Johansen, n.d.-a)

To comprehend how a suitable chain can be built, a few notions are needed. Since they can be conceived of as the probability density connected with a transition from \( x_{n-1} \) to \( x_n \), the conditional probability densities \( p(x_n|x_{n-1}) \) are frequently referred to as transition kernels. The corresponding Markov chain is referred to as time homogeneous (as its transitions always have the same distribution) if \( p(x_n|x_{n-1}) \) does not directly depend on the value of \( n \). A time homogeneous Markov chain with transition kernel \( k \), is said to have a probability density \( f \) as an invariant or stationary distribution if:

\[
\int f(x)k(y|x)dx = f(y)
\]

A Markov chain is reversible (in the sense that the statistics of the time-reversed process match those of the original process) and thus invariant with respect to a distribution if it meets the requirement known as detailed balance with respect to that distribution. The detailed balancing condition simply asserts that the likelihood of beginning at \( x \) and traveling to \( y \) is equal to the
likelihood of beginning at \( y \) and going to \( x \). (Johansen, n.d.-a) Formally speaking, given a distribution \( f \) and a kernel \( k \), one must ensure that \( f(x)k(y|x) = f(y)k(x|y) \), and with this condition, simple integration of both sides with respect to \( x \) shows invariance with respect to \( f \).

Most MCMC methods work under the assumption that if a Markov chain has an invariant distribution, \( f \), and forgets where it has been (in a proper sense), it is appropriate to use its sample route to approximate integrals with respect to \( f \). An analog of the law of big numbers can be provided by formalizing this under technical constraints. Technically speaking, this can be codified to provide an analog of the central limit theorem and the law of large numbers (commonly referred to as the ergodic theorem). The first of these results indicates that, as the no. of samples increases, we can anticipate the sample average to converge to the proper expectation with probability one; the second indicates that the estimator we obtain is asymptotically normal with a specific variance (which depends on the covariance of the samples obtained, illustrating how crucial it is that the Markov chain forgets where it has been fairly quickly). (Johansen, n.d.-a)

It might be challenging to consistently check for these criteria in practice, but they are crucial because it is simple to create examples that completely defy the rules and behave incorrectly. It is important to build Markov chains with the appropriate invariant distribution in order to utilize this method to estimate interest expectations. There are few popular methods for solving this issue, such as, Metropolis-Hastings Algorithm & Gibb’s Sampling. However, there is still an advanced method called Hamiltonion Monte Carlo (HMC), an extension of MCMC, that leverages concepts from physics to achieve efficient exploration of complex parameter spaces. Its ability to generate less correlated samples and its suitability for high-dimensional problems make it a valuable tool in various scientific disciplines. Data Augmentation of the given maintenance data in my case, has been achieved applying HMC Algorithm, using No U-Turn (NUTS) Sampler.

### 3.2.1. Hamiltonion Monte Carlo Method

A strong and effective Markov Chain Monte Carlo (MCMC) technique for sampling from intricate and high-dimensional probability distributions is Hamilton Monte Carlo (HMC). It was created to help with some of the drawbacks and difficulties that come with more conventional MCMC techniques, such as the Metropolis algorithm. Fundamental concepts from classical physics, especially Hamiltonian mechanics, are incorporated. It presents the idea of a "Hamiltonian," a mathematical function that integrates a system's kinetic and potential energies (negative log-posterior distribution). Hamilton's equations of motion control the dynamic behavior of a particle travelling in this energy environment. (Betancourt, 2017; Monte Carlo Methods-Lecture Notes, n.d.)

HMC includes an auxiliary momentum variable to help with the dynamics. A distribution, usually a conventional normal distribution, is sampled for this variable. A Hamiltonian system is defined by the combined distribution of locations and momenta, and HMC models the system's temporal evolution. The leapfrog integration algorithm is the fundamental numerical technique of HMC. It makes it possible to simulate the Hamiltonian system with accuracy and stability by discretizing the continuous dynamics into tiny steps. In order to maintain the Hamiltonian, leapfrog integration alternates between updating momenta and locations. Using
the leapfrog technique to simulate the Hamiltonian dynamics, HMC generates new states given a target distribution. After then, a Metropolis acceptance probability is used to determine whether to accept or reject the suggested state, guaranteeing precise balance and convergence to the desired distribution. (Betancourt, 2017; Johansen, n.d.-b)

### 3.2.2. No-U-Turn Sampler (NUTS) Sampler

A version of the Hamiltonian Monte Carlo (HMC) algorithm called the No-U-Turn Sampler (NUTS) is intended to automatically adjust the algorithm's important parameters, like the step size and the no. of leapfrog steps. By addressing some of the issues with manual tuning in conventional HMC, NUTS was able to improve the algorithm's adaptability and user-friendliness. The step size and no. of leapfrog steps can be manually adjusted, which can be delicate and time-consuming. However, NUTS eliminates this necessity. During the simulation, it uses a recursive method to automatically calculate the right trajectory length. (PYMC, n.d.)

The algorithm's capacity to identify the point in the trajectory where the particle makes a U-turn, suggesting that it may have fully examined the target distribution, gives rise to the moniker "No-U-Turn Sampler". When NUTS senses a U-turn, the trajectory is stopped, limiting further needless investigation. To explore the parameter space, NUTS makes use of a binary tree structure. It begins at a single location and, at each step, chooses whether to proceed left or right, so generating a binary tree of potential paths. During the sampling process, NUTS dynamically adjusts the step size and trajectory length, which enables it to effectively investigate the target distribution and adjust to its geometry. In terms of exploration, NUTS is especially effective due to its versatility and detection of U-turns, which minimizes the need for laborious manual tuning and produces useful samples. (PYMC, n.d.)

**NUTS Implementation Sequence**

- **Initialization**
  - NUTS starts by initializing a single point in the parameter space

- **Build Binary Tree**
  - It then builds a binary tree of trajectories by iteratively choosing directions (left or right) based on the current state and the gradient of the log-posterior distribution.

- **No U-Turn Criterion**
  - NUTS monitors the trajectories and stops when it detects a U-turn, preventing unnecessary exploration.

- **Update Parameters**
  - The algorithm updates the step size and trajectory length based on the information gathered during the exploration.

- **Repeat**
  - The process is repeated, and the algorithm continues to adapt its parameters and explore the target distribution until a sufficient no. of samples are obtained.
3.2.3. PYMC Bayesian Inference Library

In order to implement Hamiltonion Monte Carlo using NUTS sampler, PYMC Library on python has been utilized for model design and extraction of results for data augmentation of maintenance. Gradient-based algorithms for Markov chain Monte Carlo (MCMC) sampling, known as Hamiltonian Monte Carlo (HMC), allow inference on increasingly complex models but requires gradient information that is often not trivial to calculate. PyMC is an open source probabilistic programming framework written in Python that uses PyTensor to compute gradients via automatic differentiation, as well as compiling probabilistic programs on-the-fly to one of a suite of computational backends for increased speed. (PYMC, n.d.)

In addition, NUTS also has several self-tuning strategies for adaptively setting the tunable parameters of Hamiltonian Monte Carlo, which means you usually don’t need to have specialized knowledge about how the algorithms work.

Multiple Compatibility Issues and Solution:

A no. of compatibility issues arise when running the PYMC environment on python. After deliberate and consistent engagement in pursuit of solution, following configuration works fine with regard to inter-libraries conflicts and errors.

- Python 3.11.6
- Pandas 2.1.2
- Numpy 1.25.2
- Pymc 5.9.1
- Pytensor 2.13.1
- Arviz 0.16.1
- Matplotlib 3.8.1
- Seaborn 0.13.0
- Scipy 1.11.3

3.2.4. Some Comparison & Discussion about different MC Algorithms

1. Metropolis-Hastings:
The general-purpose MCMC algorithm Metropolis-Hastings creates a Markov chain whose stationary distribution is the desired target distribution for sampling from. MH works fairly well for low-dimensional distributions but can be inefficient in high-dimensional spaces or when dealing with correlated variables. In each iteration, it proposes a new sample based on a proposal distribution, accepts or rejects the proposed sample based on an acceptance probability, and proceeds to the next state accordingly.

2. Gibbs Sampling:
The Metropolis-Hastings algorithm's special case known as "Gibbs Sampling” was created especially for sampling from high-dimensional distributions. Gibbs sampling is a highly effective method when the joint distribution can be easily decomposed into conditional distributions. Rather than proposing a new sample for all variables at once, it updates each variable one at a time, conditioned on the current state of the other variables. In some problem
3. **Hamiltonian Monte Carlo (HMC):**

Hamiltonian dynamics concepts are used by the more advanced MCMC algorithm, HMC, to suggest new samples. By treating the state space as a manifold, it adds an extra momentum variable to the goal distribution. HMC typically has better mixing properties and faster convergence compared to simple random-walk-based methods like Metropolis-Hastings and Gibbs Sampling. It generates proposals that are more effective in exploring the target distribution, especially in high-dimensional spaces with complex geometries, by simulating the dynamics of the system using Hamilton's equations.

4. **NO U-Turn Sampler:**

To increase sampling efficiency, NUTS is an adaptation of the Hamiltonian Monte Carlo method that automatically adjusts its parameters, such as the step size and the number of leapfrog steps. Until it detects a "U-turn," which denotes that the trajectory is looping back on itself, it employs a recursive method to dynamically decide the trajectory of the Hamiltonian dynamics. NUTS makes it easier to use and more effective in practice by doing away with the requirement for manual parameter adjustment. It frequently performs better than conventional HMC, particularly when dealing with complex target distributions that have a high curvature or a large number of dimensions.

3.3. **Dataset Details and Data Pre-Processing**

The dataset that is being used in our model is available in CSV file. It comprises breakdown maintenance data of a no. of vehicles of a unique trainset/model and is available in Swedish Language. Therefore, column titles and necessary details were translated into English. There are total of seven train vehicles and maintenance data in the file has all the primary and necessary information such as vehicle no. and corresponding data related to Fault Reporting Date (FRD), Maintenance Completion Date (MCD), Fault/ Failure Category, Failure Modes, Description related to failure cause, Action Code Description etc. Since there are several categories of failures recorded in the data, methodology has been implemented only on one type of failure i.e. ‘BROMSSYSTEM’ (Brakes System). There can be many direct or indirect causes of failure in braking system, however, focus was made on failures that involved the braking pads; particularly exchange of Brake Pads. This gives a relatively clean data to be analyzed regarding only one functional failure while neglecting other problems in the braking system.

The data related to exchange or replacement of brake pads was obtained by filtering out column with the title “Åtgärdskodsbeskrivning” (Action Code Description) with ‘UTBYTT’ (Exchanged) Category. In order to make data more relevant and focused on replacement of braking pads, the column ‘Åtgärdstext’ (Maintenance Action Description) was sorted with respect to words containing replacement of brake pads. Based on these considerations, the data was processed, sorted and filtered, while rest of the irrelevant rows and columns were removed.
Since, the data augmentation is focused on the data concerning time between failures, therefore, time between failures were calculated into a separate column by finding the difference of days between two dates of the successive failures due to worn out or faulty braking pads. For each category of vehicle, data was filtered separately in order to maintain uniformity and to avoid errors due to difference of dates repetitions in every different vehicle type. Out of total 6163 maintenance jobs entries in the data for all vehicles, only 378 entries were found that were reported due to faults in braking system with faulty brake pads. A snapshot of the dataset is shown as follows:

<table>
<thead>
<tr>
<th>Vogn</th>
<th>Skadesdatum</th>
<th>Sængetidpunkt</th>
<th>Usænkigt</th>
<th>Reparatur tid</th>
<th>Time between failures</th>
<th>Skadekategori</th>
<th>Skadebeskrivelse</th>
<th>Ansænkelsesbeskrivelse</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3/07/2016</td>
<td>5/07/2015</td>
<td>5/07/2016</td>
<td>0</td>
<td>50</td>
<td>10</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>2</td>
<td>4/07/2016</td>
<td>5/07/2015</td>
<td>5/07/2016</td>
<td>0</td>
<td>45</td>
<td>9</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>3</td>
<td>5/07/2016</td>
<td>5/07/2015</td>
<td>5/07/2016</td>
<td>0</td>
<td>62</td>
<td>5</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>4</td>
<td>6/07/2016</td>
<td>5/07/2015</td>
<td>5/07/2016</td>
<td>0</td>
<td>0</td>
<td>35</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>5</td>
<td>7/07/2016</td>
<td>5/07/2015</td>
<td>5/07/2016</td>
<td>0</td>
<td>0</td>
<td>65</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>6</td>
<td>8/07/2016</td>
<td>5/07/2015</td>
<td>5/07/2016</td>
<td>0</td>
<td>0</td>
<td>35</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>7</td>
<td>9/07/2016</td>
<td>5/07/2015</td>
<td>5/07/2016</td>
<td>0</td>
<td>0</td>
<td>65</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>8</td>
<td>10/07/2016</td>
<td>11/07/2015</td>
<td>10/07/2016</td>
<td>0</td>
<td>0</td>
<td>35</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>9</td>
<td>11/07/2016</td>
<td>12/07/2015</td>
<td>11/07/2016</td>
<td>0</td>
<td>35</td>
<td>10</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>10</td>
<td>12/07/2016</td>
<td>12/07/2015</td>
<td>12/07/2016</td>
<td>0</td>
<td>1</td>
<td>39</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>11</td>
<td>13/07/2016</td>
<td>12/07/2015</td>
<td>13/07/2016</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>12</td>
<td>14/07/2016</td>
<td>15/07/2015</td>
<td>14/07/2016</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>13</td>
<td>15/07/2016</td>
<td>16/07/2015</td>
<td>15/07/2016</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>14</td>
<td>16/07/2016</td>
<td>17/07/2015</td>
<td>16/07/2016</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
<tr>
<td>15</td>
<td>17/07/2016</td>
<td>18/07/2015</td>
<td>17/07/2016</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>Børnebæg</td>
<td>Vognkontrollen</td>
</tr>
</tbody>
</table>

3.4. Time Between Failures Model

1. Estimation of most suitable distribution

Since the underlying distribution of our maintenance data is unknown, in order to identify the best fit to the data, a code was developed on python (attached in the Annexure) using ‘Distfit’ package that undertakes goodness of fit tests for fitting a probability distribution to data for multiple distributions. This compares, for any no. of classes, the observed frequency (f) to the predicted frequency (f-hat) from the model. ‘Distfit’ utilizes Sum of Squared Errors (or estimations) (SSE), also known as Residual Sum of Squares (RSS), and calculates the goodness of fit. The lowest value of RSS produced as a result of performing goodness of fit tests for a no. of probability distributions on the given data defines the most suitable fit. Upon performing such an analysis on our maintenance data, shows that the most suitable distribution of the given data is Exponential Distribution with following resultant parameters:
The exponential distribution has a probability density function (PDF) given by:

Equation 5. PDF of exponential distribution

\[ f(x|\lambda) = \lambda e^{-\lambda x} \]

- \( x \) is the random variable, \( x(0, \infty) \)
- ‘\( \lambda \)’ is the rate parameter and inverse of Scale, where scale = \( 1/\lambda \), \( \lambda > 1 \)
- Mean (Scale) = \( 1/\lambda \)
• Variance = $1/\lambda^2$

![Figure 25 PDF of Exponential Distribution with different rate parameter](PYMC, n.d.)

### 3.5. Python Implementation

To implement MCMC in Python, PyMC3 Bayesian inference library was used. The following code creates the full model with the parameters, alpha and beta, the probability, p, and the observations, observed. The step variable refers to the specific algorithm, and the sleep_trace holds all of the values of the parameters generated by the model. (Full code attached).

With Bayesian analysis, we think scale parameter as being random variable and start by creating a distribution which represents a best guess at its values. These initial values are called ‘Priors’. The resultant distribution, also referred as Proposal Distribution, of the parameter has been assumed to be Normal. Initial values have also been fed as mu(mean)=0.0 and Tau (standard deviation = 0.05). The probability of failure times has been modeled as Exponential Distribution, also called Posterior Distribution or Target Distribution, to model the observed data. Lastly, we tell PYMC to use MCMC to sample from Target Distribution. This posterior distribution incorporates our prior distribution of unknown parameter value and updates them to reflect the fact that we now have observed some data, so we must have better idea what the parameters is. (PYMC, n.d.)
Every time, NUTS sampler draws a set of random variables from proposal distribution and evaluates them against the observed data. If they do not agree with data, values are rejected and model remains in the current state. However, if the values are accepted, the values are assigned to the parameter. That’s how we get distributions of our rate of scale parameter.

Trace holds all the samples of prior distribution (for Lamda).

```python
#with pm.Model() as TBF_model:
    # Creation of the parameter for the exponential distribution
    alpha = pm.Normal('alpha', mu=0.0, tau=0.05, testval=0.0)

    # posterior distribution
    # The probability of failure times is modeled as an exponential distribution
    failure_times = pm.Exponential('failure_times', lam=1/alpha, observed=TBF_obs)

    # Draw the specified number of samples
    trace = pm.sample(2000, tune=1000)  # Adjusting the number of samples and tuning based on convergence
```

Figure 26 TBF Model for implementation of MCMC Simulation using NUTS sampler on input dataset
4. Results & Discussion

1. Trace

The python code was run initially with one MCMC Chain but it spent long time to complete the execution. Therefore, the code was run with four chains, starting simultaneously, with different starting values and utilized all the four CPU Cores of the computer. Following is the trace plot that shows the sampled values of our parameters: Shape and Scale.

![Trace Samples of Alpha (Scale Parameter) & corresponding distribution](image1.png)

The trace plots show that each state is correlated to the previous -the Markov Chain, but the values oscillate significantly – Monte Carlo sampling. The different dotted curve lines show 4 Markov chains and the graph shows that they coincided very well. Total 2000 samples were generated by trace function.

2. Posterior Plot

The posterior plots of rate parameters below, show the estimated mean value of the

![Posterior Plot of Rate Parameter with resultant mean value](image2.png)

parameter and distribution with 95% HDI.
3. Results Summary

(Estimated mean values of loc and lamda parameters)

Location = 0.0
Lamda = 0.04
Scale(1/ \(\lambda\)) = 21.174

The estimated mean value of lamda or and scale parameters are then utilized to synthesize data i.e. times between failures.

These estimated mean value of scale parameter was used to synthesize three different class of samples, consisting of 5000, 10000, 15000 data points in order to evaluate the quality of augmented data in comparison to original data as well as among these three different no, of samples. These synthetic samples were augmented to the original dataset in the form of an excel file.

Following are frequency plots showing the comparison between original dataset and synthetic dataset, with 5000, 10000 and 15000 synthetically produced data points respectively.

![Figure 29 Histograms of Original vs Synthetic Data after MCMC Simulation](image)
4. Results from Goodness of fit tests with Augmented Data

Each sample of synthetic data (i.e. 5000, 10000, 15000 data points in each sample) were augmented to the original dataset in one column in excel file, similar to the original data file. These three categories of augmented data were tested for goodness of fit tests, using the Distfit python package, for Exponential Distribution. The resultant generated value of mean (scale parameter) from each augmented data was observed and compared.

Table 1 Comparative Summary of parameters for different samples

<table>
<thead>
<tr>
<th>S.No</th>
<th>Data points in Original Dataset</th>
<th>Sample Points augmented to original dataset</th>
<th>Data points in augmented dataset</th>
<th>Estimated Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Loc</td>
</tr>
<tr>
<td>1</td>
<td>114</td>
<td>5000</td>
<td>5114</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>114</td>
<td>10,000</td>
<td>10114</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>114</td>
<td>15,000</td>
<td>15114</td>
<td>0.0</td>
</tr>
</tbody>
</table>

The parameters values from goodness of test with original dataset containing 114 times between failures were: Loc= 0.0, Scale (mean)= 25.07, Lamda (Scale Rate)= 0.039. The results after augmenting the data indicate that the mean value of time between failures from original dataset is not a precise value, given the underlying distribution be exponential distribution, and rate parameter equals to 0.039. While in our results from MCMC simulation, while increasing the no. of TBFs (augmented datapoints), the scale rate gradually decreasing with seemingly variable rate and resultant scale which is actually mean value of the each augmented data, is increasing invariably i.e. not with a constant rate. As we increase the no, of synthetic data points from 5000 to 10000, the percentage difference in scale parameter between each corresponding augmented data points, is 19.25 %. But for 10,000 to 15,000 range, it is 7.6 %; although difference of data points in both the cases is 5000 data points. This also means that scale rate is decreasing exponentially and this decreasing trend in the values of lamda and increasing trend in the values of ‘mean’ values is slowing down, thus, indicating that the value of estimated mean is approaching its true value, This is because after performing MCMC and augmenting data, the size of augmented data increases from 114 (original TBFs) to 5114, 10114, and 15114 (TBFs including original TBFs with synthetic data points) and According to the law of large numbers, the average of the results obtained from a large no. of trials should be close to the expected value and tends to become closer to the expected value as more trials are performed(Pred, n.d.).

This means that the probability of any independent outcome tends to get to its true value when samples are taken repeatedly in completely random fashion and when taken the mean, it tends to approach its real value. This also implies that variance decreases close to zero. The more samples you take, the more likely you get to the mean and MCMC does that in random fashion. This implies that randomness in Monte Carlo Simulation, as we increase the size of samples, makes our estimated mean more accurate. This has clearly been seen in our results as well. Following are the distribution plots of augmented data as we increase it from 5000 to 15000 data points.

41
Figure 30  Histograms of Augmented Data (Synthetic+original) w.r.t original data and fitted distributions with resultant parameters for 5000, 10000, 15000 data samples respectively.
5. **Future Work**

The work done in this thesis can be expanded and improvised in the following aspects:

1- The maintenance data that was considered in the MCMC Model consisted of time between failures (TBF) for failures in braking system. Braking System failures can be caused by a few no. of reasons related to different components in brake system i.e. brake pads, release cable etc. But the data was simplified to consider only broader category of failures in braking system. Therefore, the model can be evaluated for one type of fault/failure in one component.

2- The augmented maintenance data can be tested on a suitable prediction model, to predict the future times between failures and compare its results with the results produced by prediction model on original dataset. It is expected to get improved results from a predictive algorithm using the augmented dataset.

3- Since, the data consists of very small no. of dataset, the TBF model can be evaluated for different sizes of datasets to check its effectiveness and comparison of results on prediction model.

4- The given maintenance data was used to synthesize further data points and then augmented together, using MCMC NUTS Sampler. In future, the results can be evaluated using other Markov Chain sampling methods like Metropolis Algorithm.
6. Conclusion

As a result of low amount of maintenance data, data augmentation technique was developed in order to synthesize more data points and augment it to the original dataset. Markov Chain Monte Carlo (MCMC) simulation has been used by modelling the time between failures maintenance data through NUTS sampler in order to generate synthetic samples of underlying distribution parameter from target distribution (posterior exponential distribution). After finding the estimated parameter of distribution, multiple sets of synthetic data points were generated in order to augment them to the original data points that enhanced the overall dataset. Augmented datasets were then individually tested for underlying distributions with parameters for comparison. It can be concluded that with these deliberations of increasing the dataset from 5000, 10000 and then 15000 data points the results converge to show that the estimated mean of parameter of the data distribution gets closer to its true value. Further, enhancing the dataset does not produce the desirable results because originally the parameters were estimated form the observed distribution of limited data points. MCMC Simulation also shows that with increasing random datapoints, dataset reveals its estimated distribution, closer to its true distribution.

The main purpose of data augmentation of maintenance data is to utilize sufficient amount of data in maintenance prognostics for better maintenance planning of the industrial asset.
References


http://arxiv.org/abs/1701.02434


Mohammed Ben-daya, Uday Kumar, & D.N. Prabhakar Murthy. (n.d.). *Introduction to Maintenance Engineering*.

*Monte Carlo Methods-Lecture Notes*. (n.d.).


PYMC. (n.d.).


Python Code

1. **Distfit**: Goodness of Fit Test (Finding the best distribution of original data)

```python
import pandas as pd
from distfit import distfit
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

# Loading original data
file_path_original = r"C:\Users\ibrah\PythonEditorWrapper_27c659b3-cf36-4931-9173-9fb506d7255a\Data_Working_File.xlsx"

df_original = pd.read_excel(file_path_original, sheet_name="Filtered_Data")

# Extracting the data from a specific column of the dataframe that contains time between failures i.e. TBF
original_data = df_original["TBF"].dropna().values  # Convert the column to a NumPy array and remove NaN values

# Omitting negative and zero entries
original_data = original_data[original_data >= 0]

# Replace zero values with a small non-zero value (e.g., 0.00001)
original_data = np.where(original_data == 0, 0.00001, original_data)

# Fit distributions
# A series of distributions are fitted on the empirical data and for each a RSS is determined.
# The distribution with the best fit (lowest RSS) is the best-fitting distribution.

dfit = distfit(todf=True)

# Searching for the best theoretical fit on data
results = dfit.fit_transform(original_data)

# Plot distribution fit
# Make plot
# Show the plots
plt.show()
```
Python Code

```python
43.
44. dist = stats.expon
45. res = stats.fit(dist, original_data)
46. print(res)
47.
```

2. MCMC Implementation

```python
import pandas as pd
import numpy as np
import pymc as pm
import arviz as az
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import expon

def main():
    # Loading pre-processed data from an Excel file
    file_path = r'C:\Users\ibrah\PythonEditorWrapper_27c659b3-cf36-4931-9173-9f85067255a\Data_Working_File.xlsx'
    df = pd.read_excel(file_path, sheet_name='Filtered_Data')

    # Extracting the data from a specific column of the dataframe that contains time between failures i.e. TBF
    data = df['TBF'].dropna().values  # Converting the column to a NumPy array and remove NaN values

    # Omitting negative and zero entries
    data = data[data >= 0]

    #**************************Markov Chain Monte Carlo Implementation**************************

    # Defining the observed data
    TBF_obs = data

    # Creating a PyMC model
    with pm.Model() as TBF_model:
        # Creation of the parameter for the exponential distribution
        alpha = pm.Normal('alpha', mu=0.0, tau=0.05, testval=0.0)

        # posterior distribution
        # The probability of failure times is modeled as an exponential distribution
        pm.Exponential('failure_times', lam=1/alpha, observed=TBF_obs)
```

```
#Draw the specified number of samples
#trace = pm.sample(2000, tune=1000)  # Adjusting the number of samples
and tuning based on convergence

# Plot of the trace and posterior distributions
#az.plot_trace(trace)
#plt.show()

# Plot of posterior distributions of mean
#az.plot_posterior(trace, var_names=['alpha'], hdi_prob=0.95)
#plt.show()

# Print summary statistics including means
#summary = az.summary(trace, var_names=['alpha'], hdi_prob=0.95)
#print(summary)

# Extract the mean values for shape and scale from the summary and
# converting Scale_rate into Scale Parameter
# Scale Parameter = 1/Scale_rate
loc = 0
mu_mean = 21.174

# Generating synthetic data from the Exponential distribution using mean
# values of priors from priors distribution
# The following code line shall produce random data, of mentioned size,
# from exponential distribution using priors means as distribution parameters
synthetic_data = expon.rvs(loc=loc, scale=mu_mean, size=15000)

# Plot the histogram of the original data and the fitted distribution
fig, ax = plt.subplots()
ax.hist(synthetic_data, bins=30, density=True, alpha=0.5, label="Synthetic Data")
ax.hist(TBF_obs, bins=30, density=True, alpha=0.5, label="Original Data")
# Add labels to axes
plt.xlabel("Time Between Failures (TBF)")
plt.ylabel("Density")
ax.legend()
plt.show()

# Rounding the values of synthetic data to whole numbers
synthetic_data_rounded = synthetic_data.round().astype(int)

#**********Data Augmentation**********
# Creating a DataFrame with original and synthetic data merged
augmented_data = pd.DataFrame({'Combined Data': np.concatenate([TBF_obs, synthetic_data_rounded])})

# Saving the DataFrame to an Excel file
output_file = r'C:\Users\ibrah\PythonEditorWrapper_27c659b3-cf36-4931-9173-9fb506d7255a\Augmented_fileexpon.xlsx'
augmented_data.to_excel(output_file, index=False)

# Creating a DataFrame with original and synthetic data with separate columns, if required.
df_synthetic = pd.DataFrame({'Synthetic Data': synthetic_data_rounded})
output_file = r'C:\Users\ibrah\PythonEditorWrapper_27c659b3-cf36-4931-9173-9fb506d7255a\expon_synthetic_data_only.xlsx'
df_synthetic.to_excel(output_file, index=False)

if __name__ == '__main__':
    main()

3. Data Visualization through Histograms

import numpy as np
import scipy.stats as stats
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Loading augmented data
file_path_augmented = r'C:\Users\ibrah\PythonEditorWrapper_27c659b3-cf36-4931-9173-9fb506d7255a\Augmented_file.xlsx'
df_augmented = pd.read_excel(file_path_augmented, sheet_name='Augmented_data')

# Extracting the data from a specific column of the dataframe
augmented_data = df_augmented['Combined Data'].tolist()

# Removing negative values from the data
augmented_data = [x for x in augmented_data if x > 0]

# Loading original data
file_path_original = r"C:\Users\ibrah\PythonEditorWrapper_27c659b3-cf36-4931-9173-9fb506d7255a\Data _Working_File.xlsx"

df_original = pd.read_excel(file_path_original, sheet_name="Filtered_Data")

# Extracting the data from a specific column of the dataframe that contains
time between failures i.e. TBF
original_data = df_original["TBF"].dropna().values  # Convert the column to a
NumPy array and remove NaN values

# Omitting negative and zero entries
original_data = original_data[original_data > 0]

# Plot the histogram of the original data and the fitted distribution
fig, ax = plt.subplots()

ax.hist(augmented_data, bins=30, density=True, alpha=0.5, label="Augmented
Data")
ax.hist(original_data, bins=30, density=True, alpha=0.5, label="Original
Data")

# Add labels to axes
plt.xlabel("Time Between Failures (TBF)")
plt.ylabel("Density")
ax.legend()
plt.show()