BIG DATA ANALYTICS FOR PREDICTIVE SYSTEM MAINTENANCE USING MACHINE LEARNING AND ARTIFICIAL NEURAL NETWORK MODELS

By

PIUS NGWA

Registration Number: 219013772

A Dissertation Submitted in Partial Fulfilment of the Requirements of the Degree of Master of Science in Data Science

African Centre of Excellence in Data Science, College of Business and Economics

University of Rwanda

Supervisor: Innocent NGARUYE, Ph.D.

September 2020

DECLARATION

I, PIUS NGWA (Registration No: 219013772) declare that this dissertation is the result of my work compiled and presented to the African Centre of Excellence in Data Science, College of Business and Economics and has not been submitted for any other degree at the University of Rwanda or any other institution. The dissertation has been passed through the turnitin anti-plagiarism checker, found to comply with the set standards and approved by the administration.

Printed Name: Pius NGWA

Signature:

Date: 05/12/2020

Supervisor: Innocent NGARUYE, Ph.D.

Signature:

Date: 13/12/2020

ABSTRACT

In a competitive market, the desire for companies to provide quality products and/or uninterrupted services to satisfy customers and stay afloat in business is on ever increase. However, machine failure often leads to unprecedented downtime, impeding the production process and affecting businesses adversely. One big challenge companies face is to optimally utilize machine's remaining useful life while at the same time reducing downtime to “acceptable low” duration. Technological advancements have been leveraged in many ways by manufacturing industries, one of the ways being the use of sensors to capture large volumes of data representing the health state of manufacturing machines/components. The insight embodied in collected data guides the decision process for system maintenance. A prominent analysis approach that has become increasing reliable is using machine learning to mine insight from data and use it to support decision making.

In this research, we train machine learning models in python using labeled time-series data collected on production machinery, and use the trained models to predict possible machine failure the next day, thus reducing the risk exposure of employees and improving the manufacturing process. By testing our models on validation dataset, the Multilayer Perceptron neural network reliably out-performed the other models with an accuracy score of 99.99994%. This model if validated with recent data and deployed would provide inside on system failure before it happens. As a result, this will lower the cost of maintenance as planning can be done with relative ease, increase system availability since system failure becomes predictable and other measures can be taken before failure occurs, and ensure reduced customer dissatisfaction that results from long system downtime.

Keywords: Machine learning, neural networks, big data, prediction, maintenance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I ascribe this success to the Almighty God for the gift of life and knowledge. His grace and favor opened up doors for me to acquire this knowledge. He has been steadfast all along, providing me with health, strength, and needed materials to accomplish this program.

Sincere thanks to my parents NGUM Odilia and FUH NGWA Michael (of blessed memory) for the hurdles they endured leading me into the world of knowledge. They sacrificed tirelessly to set my feet on the path of knowledge acquisition and I will forever be thankful to them.

I am grateful to my family members especially my wife Nicole Manka'a, my son Fred-Brian Adahnwi, my daughters Afunwi Sandra and Abinwi Eunice for their moral, financial, spiritual support and encouragements. The sacrifice they made allowing me to stay away from home throughout my studies cannot be overemphasized.

Immense gratitude to the state of Rwanda, especially the African Center of Excellence in Data Science of the University of Rwanda for opening its doors for me to pursue an MSc degree in Data Science. Many thanks to all my lecturers especially Dr. Kabano Ignace, Dr. Christine, Dr. Eric Umuhoza, and Dr. Innocent Ngaruye who put in their contributions to guide me through my studies.

A big thank you to My Supervisor Dr. Innocent Ngaruye, who guided me through the thesis process relentlessly and took the pains to proofread this report.

DEDICATION

This work is dedicated to God Almighty for his endless love and grace upon my life.

1. INTRODUCTION

1.1 General introduction

Machines are an integral part of every economy and their impact on the daily lives of individuals cannot be overemphasized. They have become ubiquitous and are found in many sectors of life: medicine production, food processing, breweries, textile industry, aeronautics, etc. From the processing of pure mineral water, fruit juices, milk, and other dairy products, machinery plays a key role. Companies often strive to operate their machines at optimal capacity, producing quality products and/or services to satisfy their customers. However, the production chain sometimes faces some challenges which interrupt the smooth running of the machines. These include errors like the failure of a machine hardware component or the entire machine system, failure of a software program controlling the operations, human error, and at times network errors that prevent machines from communicating.

The failure of a machine due to one of the aforementioned reasons or any other reason often interrupts a process. This impedes the business unit from achieving its goals and in some circumstances, system failure poses security danger. A system failure may lead to companies incurring losses which may be high. For example, in batch processing, if a machine fails in the middle of a batch cycle, this leads to production losses as the final products become faulty. System failure also impacts customers negatively. This is because the company becomes unable to satisfy customers' demands, leading to a strained customer relationship. Some business units even lose clients in favor of competitors due to system failures. In the case of airplanes, the failure of the system can be catastrophic and even claim human lives.

In recent years, advancements in technology such as MTConnect have made it possible to capture and store large amounts of data about online machines' parameters with the help of sensors. Some operational parameters are fluid pressure, current, rotation speed, vibrations, and flow rate. Companies make use of these data to develop data-driven solutions which give them a competitive advantage in the global market. These data when analyzed, provide stakeholders with rich pieces of information like the effectiveness of the equipment, machine up-time, and capacity utilization. The analysis of these data could be used to monitor the machine's health in real-time while it is online and take appropriate action as needed.

The concept of predictive maintenance has gained increasing attention in recent years as companies strive to reduce operational costs and improve the quality of their products/services. As suggested by Figure 1.1, investments in predictive maintenance is in order to take advantage of the low maintenance cost incurred. Cambridge Advanced Learner's Dictionary (Third Edition) defines maintenance as the work needed to keep a road, building, machine, etc. in good condition. According to Traini et al. (2019), "Predictive Maintenance is a method to monitor the status of machinery to prevent expensive failures from occurring and to perform maintenance when it is required." The traditional maintenance practice has been run-to-failure. This approach fully utilizes the useful life of the machine but often leads to unwanted downtime. In predictive maintenance, one seeks to use historical data and make prognostics on when a system would fail so that corrective measures can be taken to avert the failure or at least mitigate the impact of its occurrence. It is a proactive approach that forecasts maintenance well ahead of time, reducing the downtime that normally results from system failure.

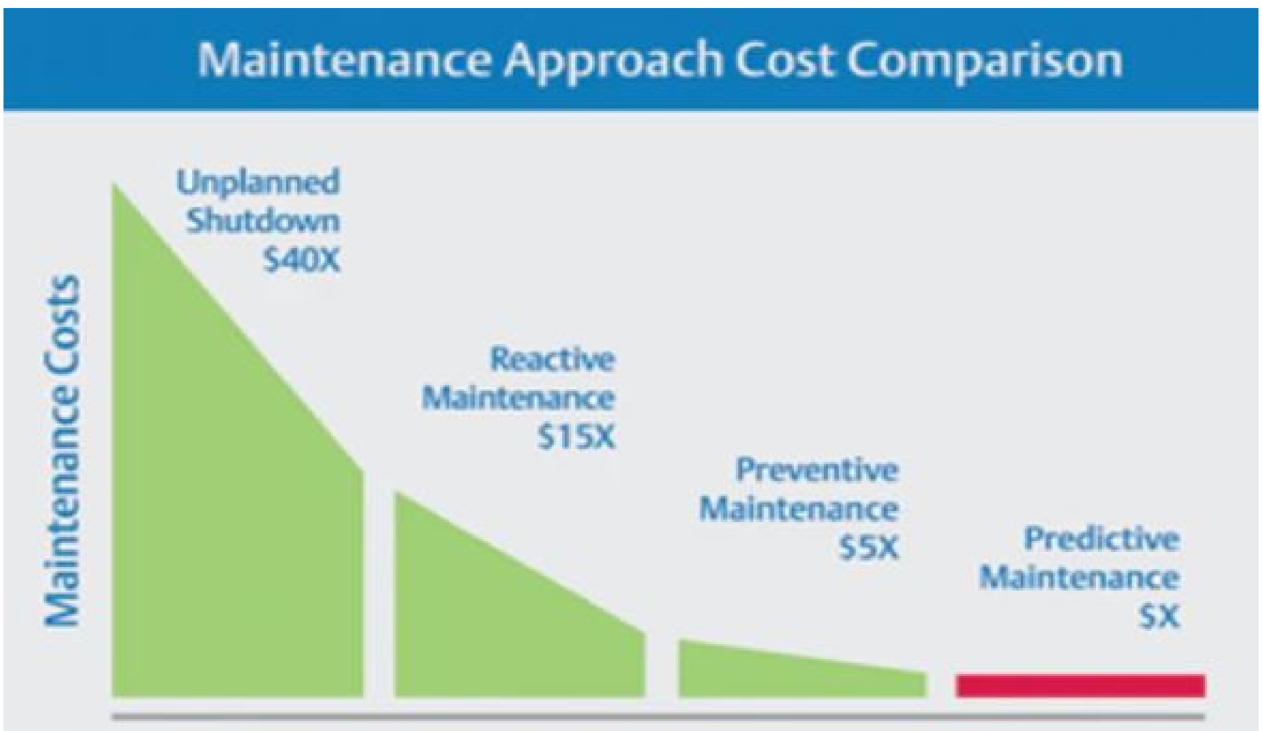

Fig. 1.1: Comparative costs of system maintenance approaches

Collected data about manufacturing machines' parameters and the errors encountered over time can be used to predict future errors. The field of predictive maintenance focuses on using such data to forecast the occurrence of an error in a system. In this research, we use data about machine failure downloaded from Kaggle (2020): an online platform with resources for training different models and competitions; to build machine learning and Artificial Neural Network algorithms that predict failure in the company's production chain.

1.2 Background of the study

The economies of most developing and developed countries rely heavily on the use of machinery for diverse purposes including production, automobile, and security. In production, for example, companies often strive to produce quality goods at optimal capacity to meet up with demand. Also, in the area of security, the maximum availability of machinery used is of utmost importance. In some cases, the machines used are very expensive. Unknown system downtime and service unavailability are often caused by the failure of a machine in use. Maintenance issues can be completely different and the predictive information to be fed to the predictive maintenance module has, in general, to be tailored to the particular problem at hand. This justifies the presence in the literature of many different approaches to predictive maintenance (Luo et al., 2013).

The simplest approach to dealing with maintenance is the Run-to-Failure (R2F) approach where maintenance interventions are performed only after the occurrence of failures. But it is also the least effective one, as the cost of interventions and the associated downtime after failure are usually much more substantial than those associated with planned corrective actions taken in advance (Susto et al., 2014). The efficient management of maintenance activities is becoming essential as it decreases the costs associated with downtime and defective products, especially in highly competitive advanced manufacturing industries such as semiconductor manufacturing (Simidu, 2001). The predictive maintenance of the industrial machine is one of the challenging applications in the new era of Industry 4.0. Thanks to the predictive capabilities offered by the emerging smart data analytics, data-driven approaches for condition monitoring are becoming widely used for early detection of anomalies on production machines (Borgi et al., 2017). Many predictive models have been proposed to mitigate hard drive failures but failure prediction in real-world operational conditions remains an open issue. Failure could be caused by unpredictable events, for example, the improper handling of a device happening occasionally in an environment maintained by an expert. However, this alone cannot explain why the high performances of failure prediction models that are found in the literature have not mitigated the problem further(Aussel et al., 2017). Machine learning-based predictive modeling is to develop machine learning-based or data-driven models to predict failures before they occur and estimate the remaining useful life or time to failure (TTF) accurately. Accurate TTF estimation plays a vital role in predictive maintenance or PHM (Prognostic and Health Management) (Yang et al., 2017). In time series analysis, it is assumed that the data (observations) consist of a systematic pattern and stochastic component; the former is deterministic, whereas the latter accounts for the random error and usually makes the pattern difficult to be identified. Previous researches usually equate stochastic component to system error and then simply discards it to not complicate the statistical analyses. However, the stochastic component potentially includes interesting and meaningful information; it must be treated with caution. It is for this reason that outlier detection becomes a hotspot research issue in recent years (Yu et al., 2014).

With the evolving new technological trends and growing system complexity, focusing on failure when designing systems for the next generation is vital. A particularly big concern is ensuring and maintaining high availability of the entire infrastructure. This is extremely important because failure to have a prior knowledge of the potential system failure might result in the following: firstly, failure of any hardware component within the infrastructure might not only result in a temporary data unavailability but in some extreme cases lead to permanent data loss. Secondly, market forces and technology trends may combine to make hardware system failures occur more frequently in the future. Thirdly, the size of hardware storage systems in modern large-scale high-performance computing infrastructures grow to an unprecedented scale with thousands of storage devices, making component failures even more difficult to detect. While there are several traditional fault tolerance techniques for dealing with and mitigating the impact of failures, there is a critical need to understand the future failure pattern and behavior of real systems(Elliott et al., 2012). Such an understanding will not only help evaluate the future failure of system component by fine-tuning the existing techniques, but will aid in the design and development of new mechanisms(Bashir et al., 2019). Predictive maintenance is often conducted by those in the industry. Most industry work is not published such as the predictive maintenance system for Verizon's cell towers(Sennaar, 2020).

While acknowledging that there is considerable literature out there on predictive maintenance, every implementation addresses the problem differently as algorithms need parameter tuning with various datasets. This study seeks to operationalize big data in real-time and translate it into actionable insights for managing production machines. We establish key performance indicators of machine failure based on a dataset downloaded from Kaggle (2020). Kaggle is a website that launches competitions for machine learning practitioners around the globe. It is a host to datasets used for building various machine learning algorithms to solve live challenges in different fields. Their datasets could be simulated or collected for a given purpose. One of the datasets is the Machine Failure data which we use in this project. In this research, the chosen machine failure data is used to build both machine learning and Artificial Neural Network models that predict machine failure. Insight from the research could be adopted and fine-tuned by any production company using their dataset and used to predict system failures in a different setting. It will also serve as a guide to other researchers who may embark on the same or similar research.

1.3 Problem statement

Industries often aim at producing or rendering services at optimal capacity. This demands that their machinery be up and running when needed. However, a major problem faced by these industries is machines breakdown often due to mechanical problems, that result in undetermined downtime. This reduces efficiency and, in most cases, companies do not meet up with their production/service set goals. This causes significant costs associated with service/product unavailability or delays in the production processes. To mitigate system downtime and unavailability which causes financial losses, stresses customer relationships among other effects, machine learning and Artificial Neural Network predictive maintenance models should be employed. These models leverage historical data to predict system failure, provide information on the type of pre-emptive maintenance required, and when it should be done. This will enhance maintenance planning which reduces system unavailability.

1.4 Project motivation

Being able to reduce system downtime is the desire of most enterprises which is as old as the concept of system automation itself. This desire has witnessed tremendous growth in recent decades partly due to competitive market nature and the ever increase in the demand of goods and services. To be successful in this competition, companies need to minimize downtime and meet customers' expectations in terms of quality of services rendered and products. Thus, the continuous strive to mitigating system downtime which interrupts production or service delivery is key while maintaining or aiming to improve product quality always. Run to failure system maintenance approach often leads to unprecedentedly high downtime. Also, when scheduled maintenance is conducted, the remaining useful life of the system is underutilized. These mentioned approaches affect the business considerably as either service/product delivery is impeded for undesirable time due to system failure or the remaining useful life of components is underutilized leading financial losses (what the system would have produced using the remaining useful life is lost when maintenance is carried out too early). Consequently, the continuous strive for a balance between the two: optimally utilize resources while at the same time bringing system downtime to an acceptable minimum (a relative term since the ideal is zero downtime). We attempt to address this issue by mining data that provides insight to stakeholders, that can guide them on when a machine can fail as they strive to optimally utilize the remaining useful life of the device.

Based on this background, the aim of this study is to select and implement algorithms which predict machine failure in a production chain.The goal is provide a tailored solution to a specific dataset and strive to achieve better performance of the models in predicting system failure. Literature suggests that an increase of 2% in performance of a model as failure prediction is concerned leads to huge sums of money being saved. Insight on failure determinants or predictors is provided and the performances of different models is evaluated. Existing research in this field of predictive maintenance is often carried on private company data and most of these research findings are not out there. This research will enhance the scheduling of system maintenance by investigating exiting approaches used and aiming to achieve better results on a given dataset. An achievement of only a 2% increase in predicting production machine failure by a model can save a company from incurring a huge loss. Predictive maintenance has a great role to play in realizing industry 4.0 goals which focus on making manufacturing faster, more efficient and more customer-centric. Christiansen (2020) notes that "manufacturers' savings from using predictive maintenance could reach between $240 and $630 billion globally by 2025". This is an enticing amount, making the endeavor worth investing in.

1.5 Objective of the study

1.5.1 General Objective

This study aims to find out the determinants of machine failures in a production chain and build an optimal system failure prediction model that predicts whether or not a system will fail the next day.

1.5.2 Specific Objectives

The specific objectives of this study are the following:

- To identify Key indicators associated with system failure from the machine failure data downloaded from Kaggle.
- To predict production machine failure using machine learning models and identify the best-performed model on the dataset.
- To compare the performance of traditional machine learning models (SVM,KNN) and Artificial Neural Network model in system failure prediction for the given dataset, identifying which performed best.

1.5.3 Research Questions

In this research, we attempt to provide answers to the following questions:

1. What are the key predictors of system failure in the machine failure dataset obtained from Kaggle?

2. To what degree of accuracy can production machine failure be predicted within a window of 1 day by the best performing model?

3. How does the performance of machine learning models in failure prediction compare with the performance of Artificial Neural Network model for the chosen dataset?

1.5.4 Significance of the Study

Machine maintenance is a practice in which every system owner is likely to invest to ensure the efficiency of his /her system. Thus, the findings of this research could be useful to companies running machines as well as in academia as it will provide findings and highlight weaknesses which future research focus on addressing. For the industry, the project will guide stakeholders' system maintenance scheduling, thus helping to mitigate unplanned system unavailability. Identifying Key Performance Indicators for system failure from the data used will serve as a pointer to what should be looked out for when considering data collected on similar systems. The strengths and weaknesses of the neural network model and other machine learning models used in the study will be highlighted. This will guide other researchers researching production system failure predictions on what to consider. To the clients of the business unit, this research will reduce the dissatisfaction they normally get due to system failure.

According to Tran (2020), manufacturers often have to deal with up to 800 hours of downtime annually. An automotive manufacturer makes about $22,000 per minute. This highlights the loss incurred by a company per minute when its operations are interrupted. A predictive maintenance system would add value to a company in many ways.

(a) Predictive maintenance eliminates the extra cost incurred from periodic or scheduled maintenance and maximizes the available time for production. To reduce the inconvenience and cost associated with system failures, some companies schedule maintenance activities periodically. This exercise comes with an extra cost which can be reduced by employing predictive maintenance techniques or a hybrid model that employs predictive maintenance in and scheduled maintenance depending on the case being considered.

(b) Machine failures can be dangerous to the operators and in some cases are deadly or cause injuries that lead to death. With predictive maintenance, workers' safety is improved as dangerous failures are reduced and become more avoidable. With prior knowledge on the timing of a machine failure, operators can take precautions and avoid being harmed by a failure if it does occur.

(c) When the predictive maintenance model is deployed in an organization, workers invest the time which would have been used to conduct excessive maintenance in performing other tasks. This may lead to an increase in the employee's output, thus benefiting the company

(d) Equipment availability and reliability is improved as predictive maintenance minimizes the number of unexpected failures.

(e) Companies strive to minimize operational costs to remain competitive in the market. By employing predictive maintenance models, a company can enforce operating cost optimization.

(f) With predictive maintenance, there is the optimal utilization of machine's remaining useful life before a maintenance act is performed. The model guides the scheduling of maintenance by providing reliable information on when the system would fail. This leads to the minimization of the remaining useful life that would be wasted to the best level allowed by the model's prediction.

1.5.5 Scope of the work

Manufacturing and processing industries aim at producing at optimal capacity. For the industries to be efficient, the machinery deployed in the production process needs to be readily available; that is to say, be in good shape and capable of being run. However, unexpected system failures which in some cases pose safety issues, do render machines unavailable and bring production to a halt. This often translates to financial losses incurred by the company and the dissatisfaction of customers as service/product delivery is interrupted. Predictive maintenance aims at mitigating system unavailability by making predictions about potential failures that could occur.

This study focuses on classification-based predictive maintenance which aims at establishing if a system being monitored is healthy or unhealthy and would fail in the nearest future. We use data on machine failure downloaded from the Kaggle website to build failure prediction models. Key Performance Indicators (KPIs) associated with the production machine's failure are determined. Machine learning models like Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN), Decision Tree, and the Recurrent Neural Network (RNN) - Multilayer-Perceptron (MLP) are developed and used to predict system failure. Model accuracy for predicting machine failure within a defined window period of 1-day is evaluated. A comparison of model performance on the dataset is made, identifying each model's strengths and weaknesses.

1.5.6 Report organization

This dissertation is structured into five chapters as follows:

- **Chapter 1 - Introduction**: This chapter introduces the general concept of predictive maintenance. The main aim of the research and its objectives are outlined alongside the benefits of such a system to its end users. The chapter clearly states the problem which we seek to find a solution for and the scope.

- **Chapter 2 - Literature Review**: In this chapter, related work on predictive maintenance is highlighted. By exploring existing solutions, insight into the performance of different algorithms used to build a predictive maintenance model has been drawn. Our choice of algorithms is guided by this knowledge acquired.

- **Chapter 3 - Implemented Solution and Methodology Adopted**: The methods employed in building the models are presented in this chapter. A description of the dataset is also done. Here, steps followed to preprocess the data, and the algorithms used to predict system failure are described.

- **Chapter 4 - Results**: This chapter discusses the achievements after embarking on the development process. The outcomes of the project developed are summarized in this chapter.

- **Chapter 5 - Conclusions and future scope**: In this last chapter, general conclusions about the predictive maintenance project are drawn. Recommendations are made on how it can be improved upon for further enhancement of the project and future works.

2. LITERATURE REVIEW

2.1 Introduction

Industry 4.0 has been greatly embraced by many companies nowadays. Companies strive to provide quality products and/or services at their optimum capacity – minimizing interruption of the supply chain as much as possible. However, system failure, is inevitable and often leads to unprecedented downtime and system unavailability. Some sources of system failure include network error, software failure, human error, and hardware failure. Leveraging big data generated daily for sound decision making has become a great tool in recent decades. As a result, predictive maintenance has witnessed enormous researches being conducted in which historical data are analysed using various approaches.

New technological trends and increasing system complexity make focus on failure when designing systems for the next generation vital. Maintaining high availability of the entire infrastructure is a big concern. The availability and reliability of a system can be improved by predicting the system component and application failures.

With industry 4.0, enormous amounts of data are captured about machines or its components with the use of sensors and this is on a rising trend. With advancements in technology, failure prediction using machine learning algorithms and neural networks has gained attention in recent years. Despite the availability of large amounts of data and a variety of powerful data analysis methods, predictive models developed for prognostic and health management (PHM) still fail to provide accurate and precise time to failure (TTF) estimates (Yu et al., 2014).

Martinez et al. (2020). used Random Forest, Long Short-Term Memory (LSTM), and Gaussian Mixture Model (GMM) to predict system failure and identify the best-performed model using time series data captured on vehicles by PACCAR. They found out that Random Forest performed the best among the used algorithms on their dataset with a recall of 43%. However, they did not perform true feature selection on their dataset with 61 features. Also, the prediction for the time window to system failure gave poor results. Kanawaday and Sane (2017) employed Autoregressive Integrated Moving Average and machine learning technologies to predict system malfunctioning that affected the product quality. Their research focuses only on predicting system failure but does not handle the prediction of time to failure of machines. Kolokas et al. (2018) used machine learning techniques to forecast fault occurrence in a production machine. Their best model could predict a fault 5 minutes to its occurrence at an accuracy less than 60%. However, they did not categorize the predicted faults to give insight as to what needs attention. Their predictions only reported if prevailing system conditions would lead to a fault or no-fault. The information about the type of fault would greatly facilitate the identification of the fault by experts and ease maintenance work. They did not equally compute moving averages for a window period (say 5 records) which could improve on the model performance.

Susto et al. (2014) used a neural network to predict the failure/anomaly of direct current motors using the motion current signature. Instead of using measured features of the DC motors, the authors employed a simulation model that generates the training data which is used to predict the motion current signature for different motor loads. These loads are grouped into five different classes and either classified as an anomaly or not. Their research yielded a 97.59% correct classification rate. However, they noted that the probabilities for all the misclassifications were borderline cases, demanding further work to improve the process.

In the study carried out by Langone et al. (2014), the authors used a non-linear autoregressive model to predict the temperature of shaft bearing in a production machine. This predicted temperature is then fed into a Least Squares Support Vector Machine to predict future alarms due to excessive heating of the bearings. The model achieved good performance up to 15 minutes (3 steps) ahead with the prediction performance reducing after 25 minutes (5 steps) ahead. Trying to predict the machine state for a complete run using the model in some cases did not yield satisfactory results. The authors also did not use other machine learning algorithms to evaluate their performances on the dataset and compare it with that of the developed model.

2.2 Definition of key concepts

2.2.1 Data mining

Data mining is a knowledge discovery process that applies appropriate algorithms to historical data in order to generate information or insight that would otherwise remain hidden. The data mining process results in the discovery of patterns in large datasets that drive the decision-making process. Data mining techniques could be used to solve different categories of problems including classification problems, regression problems, clustering problems and prediction problems. A task could either be achieved using supervised learning or unsupervised learning.

1. **a) Supervised learning:** In supervised learning, models are trained using labeled data. The goal of this problem-solving approach is to learn a mapping function that maps input variables also called predictors to an output variable – the outcome or predicted. A well-trained algorithm is one that is able to generalize on "unseen" data. That is, when given new data with only the input variables, the algorithm would accurately predict the outcome variable. To achieve this, the algorithm is repeatedly trained using a training dataset and corrected until it learns from the data to predict the outcome, producing an acceptable level of performance. Supervised learning problems could be a classification problem or a regression problem. In classification problems, the algorithm predicts a discrete value (label or class) for each input record, for example classifying an animal as either a cat or a mouse. On the other hand, a regression problem predicts output on a continuous scale. An example is predicting the height of an individual. A good model should remain accurate over time until the nature of the data changes. In which case, the model needs to be retrained to adapt to the new changes.

2. **b) Unsupervised learning:** In this type of problems, the algorithm is fed with unlabelled data. The training dataset does not contain information about the desired outcome or correct answer. It is allowed to learn patterns found in the presented data on its own. A great deal of available machine data is unlabelled. Unsupervised learning is used to solve problems including clustering, association and anomaly detection.

The problem addressed in this thesis is a classification problem. We employ supervised learning techniques to achieve desired outcomes.

2.2.2 Predictive maintenance

Predictive maintenance is a technique to forecast the future failure point of a machine component, so that the component can be replaced, based on a plan, just before it fails. This helps to minimize equipment downtime and to maximize the component lifetime(RENOVETEC, 2020). With predictive maintenance, the status of a system is continuously observed and an alarm raised before failure occurs. This allows system owners to take appropriate action, preventing or minimizing unexpected failure.

According to Christiansen (2020) "predictive maintenance is a proactive maintenance strategy that tries to predict when a piece of equipment might fail so that maintenance work can be performed just before that happens". The predictions are based on the condition of the equipment that is evaluated using data gathered captured with various condition monitoring sensors and techniques.

Researches that aim to develop predictive maintenance problems employ one of several approaches. One of the approaches addresses this as a regression problem in which a model is built to predict the number of cycles left before a machine in-service will fail. Another approach focuses on developing a multiclass classifier. The model is used to predict if a machine will fail within a given window period t or within a period $[t_0, t_2]$ or if the machine will not fail at all over the entire period t_0 to t_2, t_0 being the start of the period and t_2 being the horizon ahead. The third approach is to develop a model that performs binary classification. This model predicts if a machine will fail within a given window period t or not. This thesis used the third approach and developed models that predict machine failure.

2.2.3 Time Series Data

In this thesis, we analyse time-series data from a manufacturing company and use machine learning and Artificial Neural Network techniques to predict machine failures. Time series is data with a sequence structure with a time dimension that can be discrete or continuous. Also, the values recorded can either be discrete or continuous. A time series $X = \{x(t)|1 \leq t \leq m\}$ is a sequence of d-dimensional observations vector $x(t) = \big(x_1(t), x_2(t), ..., x_d(t)\big)$ ordered in time.

Mostly these observations are collected at equally spaced, discrete time intervals. It is called a univariate (or single) time series when d is equal to 1 and a multivariate time series when d is equal to or greater than 2(Yang and Shahabi, 2004). Time-series data consist of data points recorded over equally spaced time intervals. The time could be in any measuring unit of time such as second, minute, hour, day, week, month, etc. Time-series data can be **stationary** or **nonstationary** and its structure is better understood through time-series analysis. A stochastic process $\phi(t)$ is said to be strictly stationary if the joint distribution of $\phi_{t_1}, \phi_{t_2}, ..., \phi_{t_n}$ is the same as the joint distribution of $\phi_{t_1+k}, \phi_{t_2+k}, ..., \phi_{t_n+k}$ for all $n, t_1, ..., t_n$ and k. In other words, shifting the time origin by an amount k does not affect the joint distribution because it depends only on the intervals between $t_1, ..., t_n$ called **lag**. The bias for such a distribution is zero. Time series data can also be said to be **"weakly stationary"** or **"second-order stationary"**. A process is called **"second-order (weakly) stationary"** if its mean is constant and its auto-correlation coefficient (ACF) depends only on the lag, so that

$$E(\phi(t) = \mu \quad \text{and} \quad Cov(\phi(t), \phi(t+\tau)) = \gamma(\tau).$$

No assumptions are made about the higher moments than those of the second order. By letting $\tau = 0$, the variance is implied to be constant. Here, τ is the time lag and γ is the autocovariance function(Robinson, 2009).

2.2.4 Data leakage

Data leakage is when information from outside the dataset is used to create a machine learning model. This information allows the model to learn something that is normally not found in unseen data, leading to a drastic change in model performance. The model behaves like one that is overfitted. Similar to the concept of overfitting, data leakage leads to models that appear performant on the training data but do not generalize on unseen data at inference time(Yang and Shahabi, 2004; Robinson, 2009). Data leakage could be caused by a feature that is present in the training dataset but absent in the test dataset. It could also be as a result of duplicate records found in both the training and the test dataset. That is, the training model sees the information that it will, later on, need to predict in the test dataset. This is a consequence of poor splitting of the data into training and test sets. By using features whose values change over time, data leakage can also occur. This often happens when data used for model construction is retrieved directly from a database. A feature may be updated with time and this would affect the behavior of the model drastically.

Data leakage is a concept that needs to be taken into consideration when dealing with predictive maintenance problems given that most of the data are unbalanced and need to be resampled. In preprocessing our data, we apply general guidelines for mitigating data leakage to ensure our models were not overly optimistic.

2.2.5 Data imbalance

Imbalanced data is talked about in a classification problem when the number of observations per class in not equally distributed. A class referred to as the majority class often has a large number of observations while the minority class(es) has/have a much fewer number of observations. With data imbalance, many machine learning algorithms are subject to a frequency bias during training in which they lay more emphasis on learning from observations in the majority class(Luigi, 2020; Jordan, 2020; Choudhury, 2020).

3. METHODOLOGY

3.1 Approach

Data mining is a field that applies various techniques to extract knowledge or insight from large volumes of data. Data mining applies different paradigms to analyze secondary data and extract information or/and make predictions. By developing models, the knowledge gained by them through the training process could be applied to perform analysis of streaming data and take informed, data-driven actions. Data mining models can be classified into two categories: predictive model – which studies patterns in historical data and can make predictions; and descriptive model - which provides summary statistics of the data. Machine Learning applies different algorithms: neural networks, clustering techniques, and tree-based algorithms to mine knowledge from data. In this research, we employ machine learning and Artificial Neural Network techniques to address the machine failure prediction problem. To achieve this goal, our actions are guided by applying the following steps: data preprocessing, model selection based on research on existing methods and metrics, model training and validation, and testing and evaluating model performance scrupulously. The algorithms are optimized by conducting parameter tuning for best performance. This process is followed by result interpretation with the final step being using the interpreted results for further action.

3.1.1 Data Collection and preprocessing

Data used for this research is secondary data. Historical time-series data that contain 24 different parameters of machine components(23 potential predictors and 1 outcome), measured in time-slots of one day from December 2008 to June 2017 is obtained from Kaggle.com. Obtained data is messy containing 16 numeric and 4 categorical attributes that have null values. For numeric attributes, the mean values for the respective features having nulls have been used to replace the null values while for the categorical features ($Parameter1_Dir$, $Failure_today$, $Parameter2_9am$ and $Parameter2_3pm$) null values are replaced with the respective modes of the features.

In order to achieve one of the objectives which is to identify Key features that determine machine failure, feature analysis is conducted to determine a reduced set of features that best captures the variance in the data. Using Scikit-learn's Principal Component Analysis (PCA) module, we identify that 17 of the 23 predictors capture almost 100% of the variance in the independent variable as shown in Figures 3.1a and 3.1b. That is to say, about 100% of the contribution of the independent variables to the dependent variable is captured by 17 features. This guided us on the number of features to consider. Using three algorithms: $ExtraTreesClassifier$ from Scikit-learn's ensemble package; SelectKBest algorithm with $f_classif$ score function, and Recursive Feature Elimination all from Scikit-learn's feature selection package; 13 features from the independent variables of the dataset (capturing about 94% of total variance in the data) were selected based on a majority vote from the three algorithms. The features Max_Temp, $Leakage$, $Electricity$, $Parameter1\_Speed$, $Parameter4\_9am$, $Parameter4\_3pm$, $Parameter5\_9am$, $Parameter5\_3pm$, $Parameter6\_9am$, $Parameter6\_3pm$, $Parameter7\_3pm$, $Failure\_today$, $RISK\_MM$ were identified as the first 13 features with higher influence on the dependent variables as voted by the three algorithms used for feature selection. Although the PCA showed that 17 features captured almost 100% of the variance in the data, we discovered after performing seasonal decomposition of the data as shown in Figure 3.3, that most of the features were noisy(the residual component in orange color is high). In the Figure 3.2, the top layer in blue represents the actually observed measurement, the green shows the Trend, that is how that values change over time, the red shows seasonal fluctuations of the values and the orange is the noise or residual component. As a result of the presence of noise in the data, we further reduced the dimension to reduce part of the noise affecting the model's performance.,

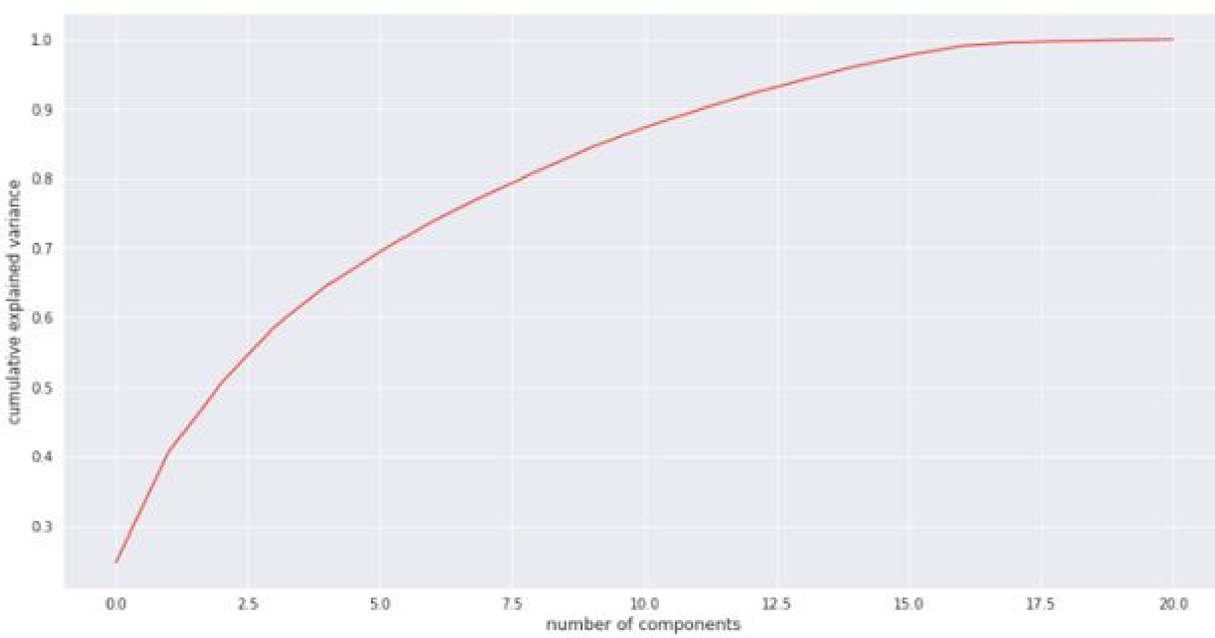

Fig. 3.1: Cumulative explained variance vs number of features

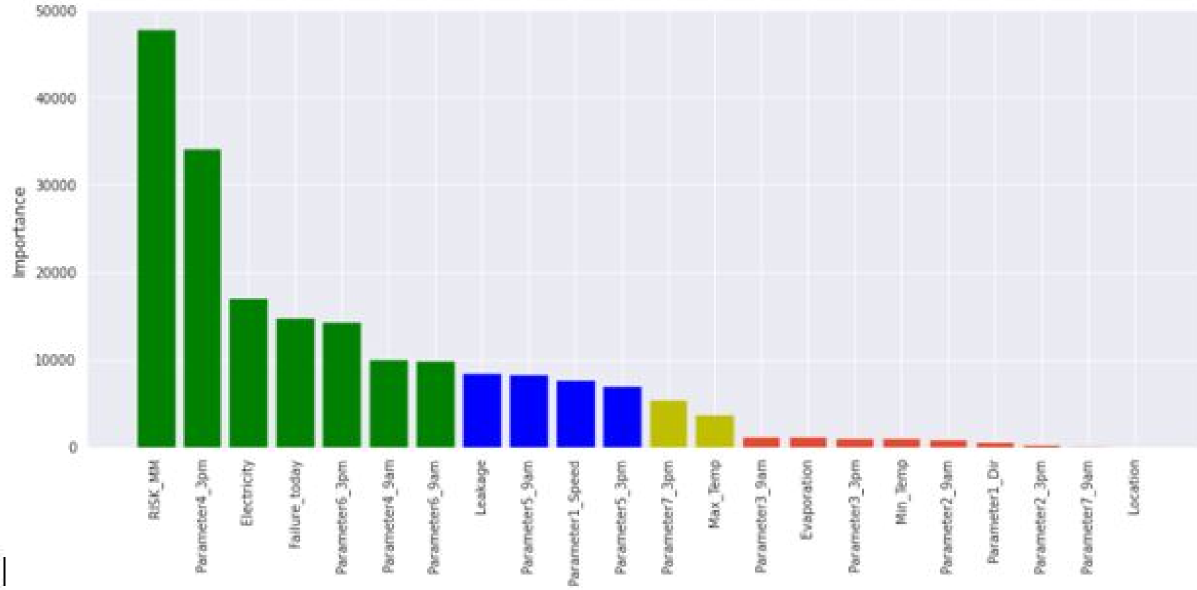

Fig. 3.2: Bar chart showing dataset feature importance

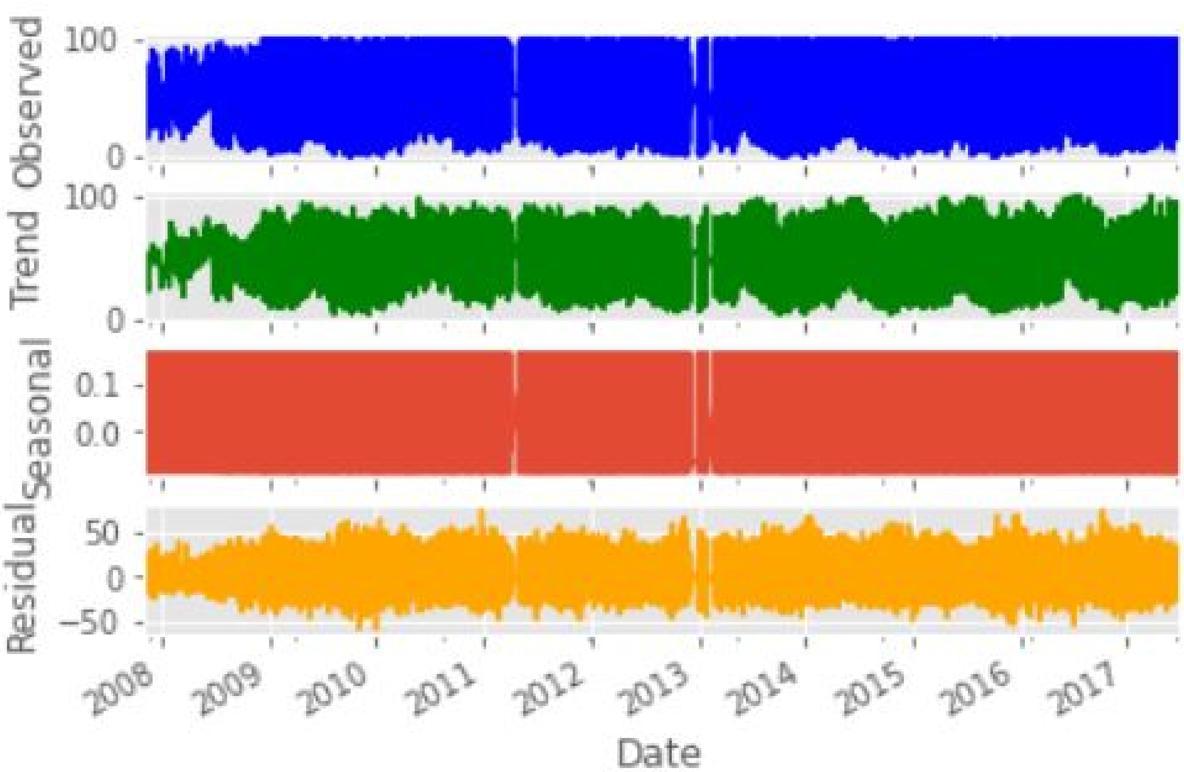

Fig. 3.3: Seasonal decomposition of $Parameter4_3pm$

Feature selection has advantages which include:

(i) By identifying and using only fewer features, the algorithm used trains faster since there is less data to handle.

(ii) Feature selection reduces redundancy from the data by removing some noise. This reduces the chances of model overfitting.

(iii) The accuracy of the model might increase since some misleading data is removed.

3.1.2 Applicable Algorithms

Machine learning models will be trained with the dataset and used to predict if the system will fail the following day or not. The supervised machine learning models: Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN) and Decision Tree classifier are used to predict system failure and their performances compared. The performances of these models are also compared with the performance of the Multilayer perceptron model on the dataset. In model evaluation, we consider the metrics Area Under the Curve (AUC), accuracy, precision and recall of the models. Since we are dealing with imbalance data, the key metric being considered is recall. Recall will give us a fraction of the relevant instances of machine failure that are correctly predicted by the algorithm. This is key as a large value of accuracy resulting from correctly predicting the majority class (normal state of machine) can mislead us to think the model is doing a great job whereas it is not. We use gird search cross validation to perform parameter tuning in order to improve on model performance and report best parameters obtained.

3.1.3 Classification

A classification problem seeks to categorize data into predefined classes or groups. A classification algorithm learns from a given dataset and when a new data point is fed into the trained algorithm, it classifies it into one of the classes based on similarities. During the training process, a function is established which maps data points to a discrete output variable.

3.1.4 Random Forests Classifier

According to Cutler et al. (2007), a Random Forest is a tree-based ensemble with each tree depending on a collection of random variables. More formally, for a p-dimensional random vector $X = (X_1, X_2, ..., X_p)^T$ representing the real-valued input or predictor variables and a random variable Y representing the real-valued response, we assume an unknown joint distribution $P_{XY}(X, Y)$. The goal is to find a prediction function $f(X)$ for predicting Y. The prediction function is determined by a loss function $L(Y, f(X))$ and defined to minimize the expected value of the loss $E_{XY}(L(Y, f(X)))$ where the subscripts denote the expectation with respect to the joint distribution of X and Y.

Intuitively, $L(Y, f(X))$ is a measure of how close $f(X)$ is to Y; it penalizes values of $f(X)$ that are a long way from Y. Typical choices of L are *squared error loss*:

$$L(Y, f(X)) = (Y - f(X))^2 \quad \text{for regression and } \textit{zero-one}\text{: loss for classification}$$

$$L(Y, f(X)) = I(Y \neq f(X)) = \begin{cases} 0 & if\ Y = f(X) \\ \\ 1 & otherwise \end{cases}$$

A random forest classifier performs recursive binary partitioning of the predictor space by applying a sequence of binary splits on each predictor. The "root" node of a Random Forest comprises of all the predictor space. "Leaf nodes" refer to nodes on the tree that are not split. They terminate the partitioning process with each leaf node belonging to a particular class.

3.1.5 Support Vector Machine (SVM)

A support Vector Machine is a supervised learning model widely used in the discipline of pattern recognition for classification purposes. A classification problem often seeks to obtain a decision surface or line that separates data points into different classes. The best **separating hyperplane** is that which separates the data points into classes with the maximum margin around the separating hyperplane. Data points which lie closest to the hyperplane are called Support Vectors. A **Support Vector** Machine is an algorithm that efficiently learns in a high dimensional space and finds an optimal solution for the positioning of the separating hyperplanes/line. In a classification problem, during the training process, an SVM produces a discriminant function that can be used to predict labels for new data records. Awad and Khanna (2015) state that SVM offers a principled approach to machine learning problems because of its mathematical foundation in statistical learning theory. SVM constructs its solution in terms of a subset of the training input. It has been extensively used for classification, regression, novelty detection tasks, and feature reduction. SVM stores all the training data in memory during the training process until it finds the optimal parameters. Once these optimal parameters are found, the SVM only needs the Support Vectors when predicting new observations. It has reduced computational time for testing and storage.

3.1.6 Multilayer Perceptron (MLP)

Multi-Layer Perceptron (MLP) is a type of feed-forward neural network. It consists of a series of interconnected nodes called neurons which are grouped into layers. Each neuron can receive multiple inputs from the previous layer, and each input has a weight. The sum of the weighted inputs is modified by a non-linear function, called the activation function, which allows the MLP to approximate non-linear functions. An MLP does not make any prior assumptions about the distribution of the data(Gardner and Dorling, 1998).

3.1.7 Decision Tree Classifier

Decision tree is a supervised machine learning approach used to solve both regression and classification problems although its greater use is for classification. The structure of a decision tree classifier consists of a root node, internal nodes which represent the parameters/features of the dataset having two or more branches which represent the decision rules, and the leaves which are the decisions or final outcomes. The best attributes for selecting the root node and sub-nodes are chosen using a technique called attribute selection measure (ASM). Two of the most used ASMs are Gini Index and Information Gain. Gini index measures the impurity or purity of a dataset attribute. An attribute with a low Gini index is good for use to split a decision tree. Information gain calculates how much information an attribute gives concerning a class. It measures changes in entropy after a dataset is segmented based on an attribute. Decision tree algorithm always tries to maximize the value of the information gain and a node or attribute with highest information gain is split first(Javatpoint, 2020).

3.1.8 K-Nearest Neighbors (KNN) Classifier

A KNN classifier is a supervised machine learning technique that considers the similarities between a new case and existing categories and puts the new case into the class to which it is most similar. This algorithm is memory consuming as it stores all the available categories into memory during the training phase. It is a non-parametric algorithm and does not make any assumptions about the underlying dataset. To classify a new data point, KNN calculates the Euclidean distance of the K-nearest neighbors to the new point. It then assigns the new data point to the class to which the highest number of neighboring data points belong.

3.1.9 Evaluation Metrics

Machine Learning models have different metrics used to evaluate how well they achieve the tasks they are built to perform. Some of these metrics are precision, recall, accuracy, Receiver Operating Characteristic (ROC) curve, and Area Under the Curve AUC. The evaluation metrics recall, precision score, ROC-AUC, confusion matrix were used to determine the performance of developed models on unseen data. Considering the fact that the data was imbalanced, we used precision-recall curve since they are appropriate for evaluating model performances in the case of imbalanced data. Recall will give us a fraction of the relevant instances of machine failure that are correctly predicted by the algorithm, and therefore tell us how well the algorithm is predicting the failure instances. F-Measure or F_1 score calculates the harmonic mean of the precision and recall. It is considered harmonic mean because both precision and recall are rates. While F_1 score measures model performance for a specific threshold, the **area under a precision-recall curve** summarizes the performance of a model over a set of threshold values.

The **confusion matrix** is used to measure the performance of a classification model where the output belongs to two or more classes. It represents prediction output in a form showing the true/actual values against the predicted outcome. Information gotten from a confusion matrix can be used to compute recall, precision, sensitivity, accuracy, and ROC-AUC. The elements of a confusion matrix are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

- **True Positive (TP)**: A prediction is called True Positive if the predicted value is true and the actual value is true. For example, predicting that a machine fails for a given observation and it turns out that the state of the machine for that record is true. Another example would be correctly predicting that a woman is pregnant.

- **False Positive (FP)**: is a *Type 1 Error* committed by the prediction model. The model predicts value as positive for a data record but the actual value corresponding to the record is negative. That is, it predicted positive and it is false. For example, a model predicts that a machine failed of which the machine did not fail.

- **False Negative (FN)**: It is also referred to as *Type 2 Error*. The predicted value is negative but the actual value is positive. For example, incorrectly predicting that a machine will not fail. Another example is predicting that a woman is not pregnant but she is.

- **True Negative (TN)**: Predicted values as negative and turns out to be negative. For example, predicting correctly that a machine will not fail.

Recall: It measures how many of the true positives were found (recalled). That is, out of the records belonging to the positive classes, how many were correctly predicted. In predictive maintenance, recall is a good measure of model performance since the data is always unbalanced. The higher the recall, the better the model performance. Recall is computed using the following formula.

$$Recall = \frac{TP}{TP + FN}$$

Precision: It measures what fraction of the returned hits were true positive. It addresses the question of what proportion of positively classified instances was correct. The formula for calculating precision is as follows.

$$Precision = \frac{TP}{TP + FP}$$

Accuracy: It is the fraction of prediction that the classification model got correctly. It can be calculated using the mathematical formula:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{TP + TN}{TP + TN + FP + FN}$$

ROC Curve and AUC

A ROC curve is a graph showing the performance of a classification model at different thresholds. It plots the True Positive Rate(TPR) against the False Positive Rate(FPR). TPR is computed using the formula for computing recall.

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate: is calculated mathematically as

$$FPR = \frac{FP}{FP + TN}$$

Area Under the Receiver Operating Characteristic Curve (AUC) measures the entire area under the ROC graph and provides an aggregate measure of performance of the model for all the classification thresholds. The value of AUC lies between zero and one. The greater the AUC value is than 0.5, the better the performance of the classification model. An AUC value of 1 depicts a "perfect" model. The closer the AUC value to 0, the poorer the performance on the model is. A model with an AUC value equals 0.5 has no class separation capability. That is, its discrimination capacity to differentiate between the positive class and the negative class(es). An AUC value of 0 implies the model predicted all data points belonging to the negative class as positive and all those belonging to the positive class as negative.

3.2 Exploratory Data Analysis

The dataset used to conduct this research was obtained from the website bigml.com(Bigml, 2020). The sample machine failure data consists of machine parameters recorded every day from the 1st of December 2008 to 24th June 2017. It has a total of 142193 observations and 24 attributes. Sensors were used to collect different measurements related to the machine's state as it operated. The original dataset consists of five categorical variables: $Parameter1\_Dir$, $Parameter2\_9am$, $Parameter2\_3pm$, $Failure\_today$, $Failure\_tomorrow$. The target variable of interest is "$Failure\_tomorrow$". This feature is categorical with "Yes" corresponding to observations during which the machine failed and "No" corresponding to observations where the machine was online and running normally. The data had no additional information describing the features. Table (3.1) presents the features and their datatypes.

Tab. 3.1: Description of data features

S/N	Feature	Description	Data Type	Number of Nulls
1	Date	A date-time feature showing the date and the time at which the data observation was recorded	Date	0
2	*Min_Temp*	The minimum temperature of the system recorded for a given day	Numeric	637
3	*Max_Temp*	The maximum temperature of the system recorded for a given day	Numeric	322
4	Location	The location of the system	Categorical	0
5	Leakage	Machine fluid leakage	Numerical	1406
6	Evaporation	Fluid evaporation	Numerical	60842
7	Electricity	Numerical value of the machine's electricity used	Numeric	67816
8	*Parameter1_Dir*	This recorded the direction of a moving part of the machine	Numeric	9330
9	*Parameter1_Speed*	Recorded the speed of the part	Numeric	9270
10	*Parameter2_9am*	A categorical feature read every at 9am using a sensor	Categorical	10013
11	*Parameter2_3pm*	A categorical feature read every at 3pm using a sensor	Categorical	3778
12	*Parameter3_9am*	A numerical feature read every at 9am using a sensor	Numeric	1348
13	*Parameter3_3pm*	A numerical feature read every at 3pm using a sensor	Numeric	2630
14	*Parameter4_9am*	A numerical feature read every at 9am using a sensor	Numeric	1774
15	*Parameter4_3pm*	A numerical feature read every at 3pm using a sensor	Numeric	3610

16	*Parameter5_9am*	A numerical feature read every at 9am using a sensor	Numeric	14014
17	*Parameter5_3pm*	A numerical feature read every at 3pm using a sensor	Numeric	13981
18	*Parameter6_9am*	A numerical feature read every at 9am using a sensor	Numeric	53657
19	*Parameter6_3pm*	A numerical feature read every at 3pm using a sensor	Numeric	57094
20	*Parameter7_9am*	A numerical feature read every at 9am using a sensor	Numeric	904
21	*Parameter7_3pm*	A numerical feature read every at 3pm using a sensor	Numeric	2726
22	*Faillure_today*	A categorical feature with two values "Yes" if system failed today and "No" otherwise	categorical	1406
23	*Risk_MM*	A numeric value attribute specifying the risk associated to system failure	Numeric	0
24	*Failure_tomorrow*	A categorical variable that indicates if a record of machine attributes corresponds to the machine failing the next day or Not."Yes" for machine will fail and "No" to mean normal state of the machine	Categorical	0

The data was imbalanced with the number of healthy operation records far greater than faulty records. Only 31877 records of the entire dataset had observations value"Yes" for failure, making 22% of the entire records. Data imbalance often harms the model's prediction accuracy and generalization performance. The target variable "*Failure_tomorrow*" was converted to a binary with 1 representing "Yes" and 0 representing "No". To raise an alarm before the occurrence of a failure, models developed would predict if a machine will fail the next day or not. This future prediction would make system operators to be aware that the system is about to fail soon.

The original dataset contained missing values as shown in Table 3.1. After careful observation of the data, we replaced missing values for numerical features with their respective mean values. For null values of the categorical features, we used the respective modes.

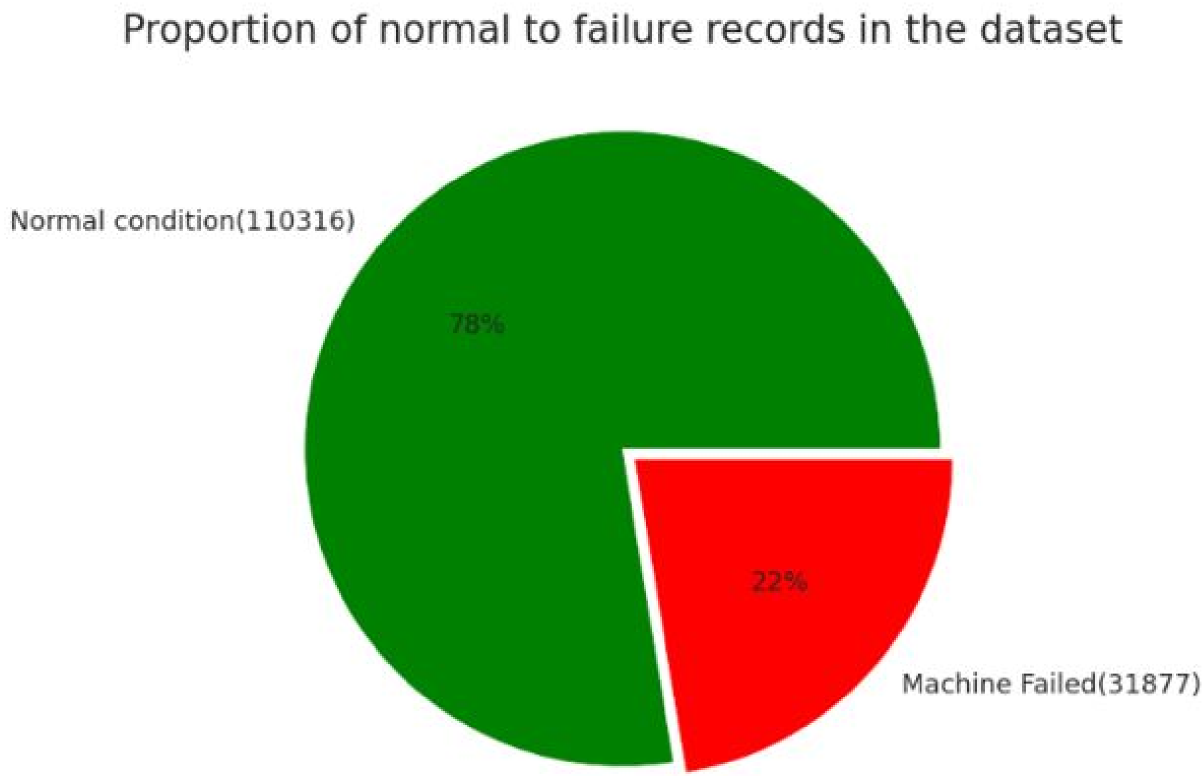

Fig. 3.4: Percentage representation of records corresponding to normal machine working conditions and failure condition

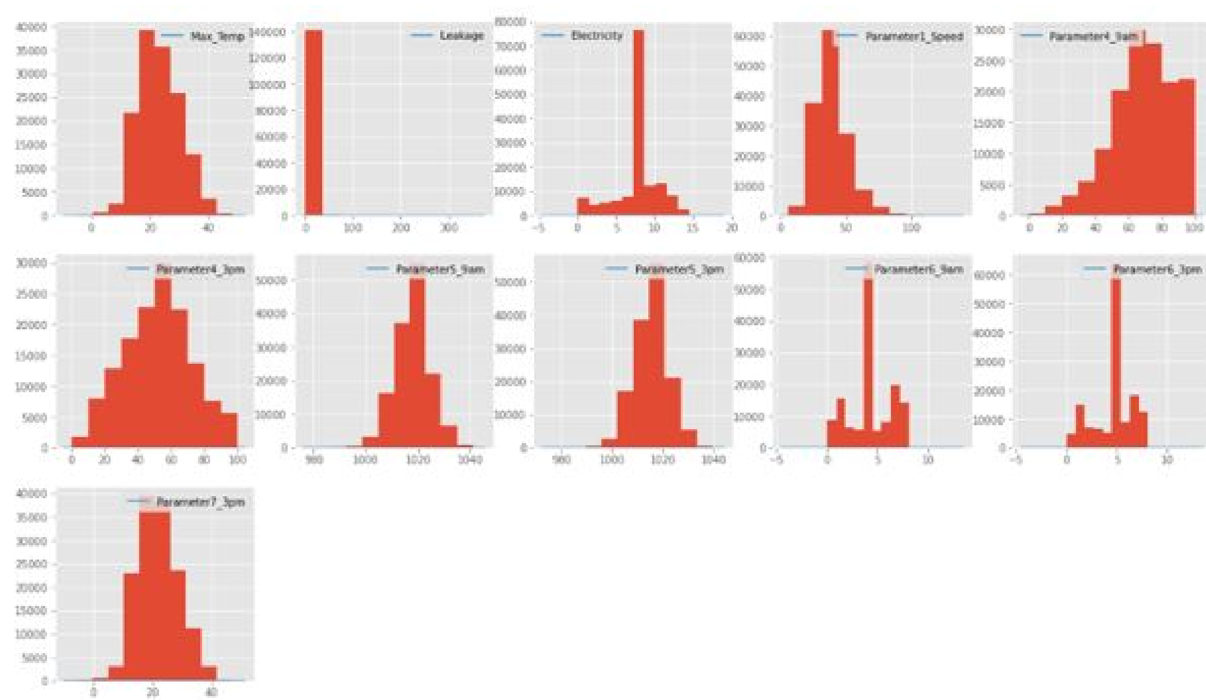

Fig. 3.5: Distribution of the different variables

3.2.1 Underlying assumptions

In our analysis, we made the following assumptions:

- A basic assumption in time series analysis is that some aspects of the past pattern will continue to remain in the future.
- Also under this setup, often the time series process is assumed to be based on past values of the main variable but not on explanatory variables which may affect the variable/system.
- In the time series analysis, it is assumed that the data (observations) consist of a systematic pattern and stochastic component; the former is deterministic, whereas the latter accounts for the random error and usually makes the pattern difficult to be identified.

3.3 Programming technology and programming tools used

To carry out data analysis, the following programming tools have been used

- Jupyter Notebook running on Python 3.7.2
- Pandas '0.19.2' and Numpy '1.11.2' for data wrangling and analysis
- Matplotlib '1.5.3' and Seaborn '0.7.1' for visualization
- Scikit-learn '0.18.1' for machine learning algorithms

4. MAIN RESULTS

This chapter summarizes the results achieved by using the dataset to train and classify the health state of the machines using both machine learning and Artificial Neural Network models. We trained 5 machine learning models and used them to predict if a machine will fail in one day or not given its current conditions. We employed five machine learning algorithms: Decision tree classifier, Random Forest Classifier, K-Nearest Neighbours classifier, Support Vector Machine and one neural network – Multilayer perceptron to perform the classification tasks.

The dataset used comprised of 142193 observations/records in total collect each day from December 1st 2008 to 24th June, 2017. In the process of model development, the data was split into three parts: training set which made up 60% of the entire dataset (85315 records), 20% for the validation set (28439 records) and the test set comprising 20% of the data (28439 records). Since we are dealing with predictive maintenance problem, following recommended practices found in literature, the first 85315 records of the time-series data were taken to be the training set, the next 28439 records as the validation set, and the last segment was taken for the test set. This is so because wear and tear of machines occur over time and the characteristics change as the machine degrades. Thus, a machine's state today is closely related to yesterday's state than its state five months ago. By splitting the data in this manner, the degradation pattern is easier for a model to learn as oppose to a situation where the dataset is split randomly. The training dataset was used to train and fine-tune the different models. Each model's validity was checked by using it to predict the outcome of the machine using the validation dataset and then confirming the performance on the test dataset. It is important to note that due difficulties running the algorithms on limited infrastructure, parameter tuning using GridSearchCV algorithm was not conducted as desired. Several attempts witnessed models running continuously for over 24 hours if internet connectivity did not fail. Thus, we reduced the set of values used during parameter tuning in order to arrive at some results. The trained models and their performances on the 'unseen' dataset are summarized in the section that follows. It is worthy to note that for tree-based algorithms used (Decision Tree classifier and Random Forest Classifier), the models recorded quite a different behavior as predicting machine failure is concerned. Of the 13 predictors chosen, one is "RISK_MM". When we used the dataset containing this variable as part of the predictors, both the Decision Tree and Random Forest Classifiers recorded 100% accuracy on the validation and test datasets. But once the "RISK_MM" variable was removed from the predictors, these two algorithms did not perform well, recording a recall of only about 45%.

4.1 Decision Tree Classifier

Bellow is the summary of performance metrics for the Decision Tree Classifier

Tab. 4.1: Summary of performance metrics for the Decision Tree Classifier

Model Name	**Machine Condition**	**Precision**	**Recall**	F_1**_Score**	**Accuracy**	**AUC**
Decision Tree Classifier	Failed (1)	1.0	1.0	1.0	100%	1.0
	Normal (0)	1.0	1.0	1.0		

The decision tree model after training on the train dataset recorded an accuracy of 100% in prediction for the unseen dataset for both validation and test cases. Table (**??**) summarizes the metrics of the classifier on the unseen or test dataset. This shows that the model generalizes well on unseen data. Figure (4.1) shows the confusion matrix of the decision tree classifier. All the records belonging to normal operating conditions (records in the test data with value 0) for the machine were correctly identified and predicted as well as the records during which the machine witnessed a fault (with value 1). Figure (4.2) shows a plot of the Receiver Operating Characteristic (ROC) curve the area under the curve for the decision tree classifier. As expected, the area under the curve (AUC) equals 1. One metric of great importance when solving a classification problem using imbalance data is the recall. In Figure (4.3), the precision-recall curve for the decision tree classifier is shown.

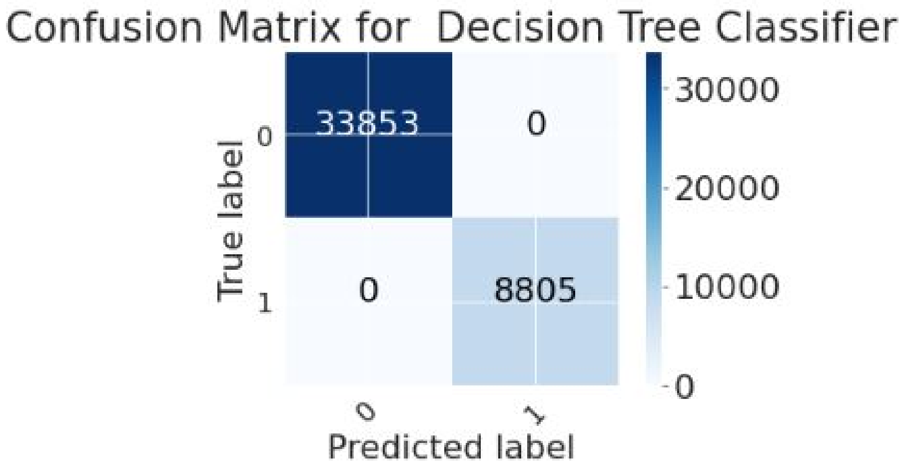

Fig. 4.1: Confusion matrix for decision tree classifier

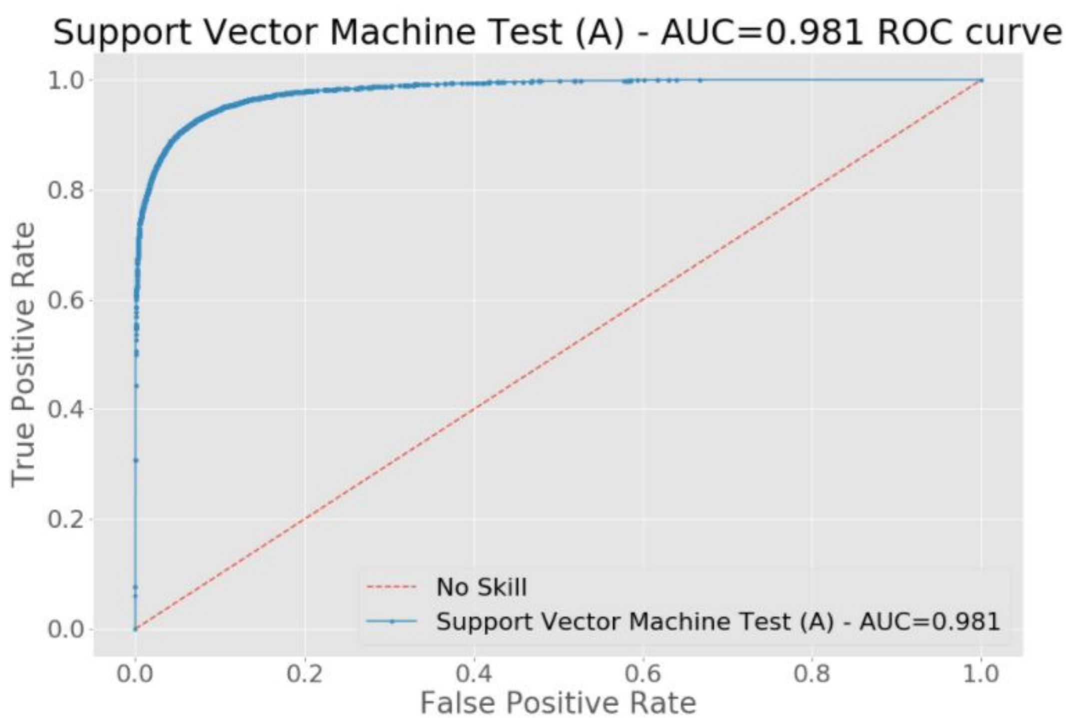

Fig. 4.2: Receiver Operating Characteristic Curve and Area Under the Curve (ROC-AUC) for decision tree classifier

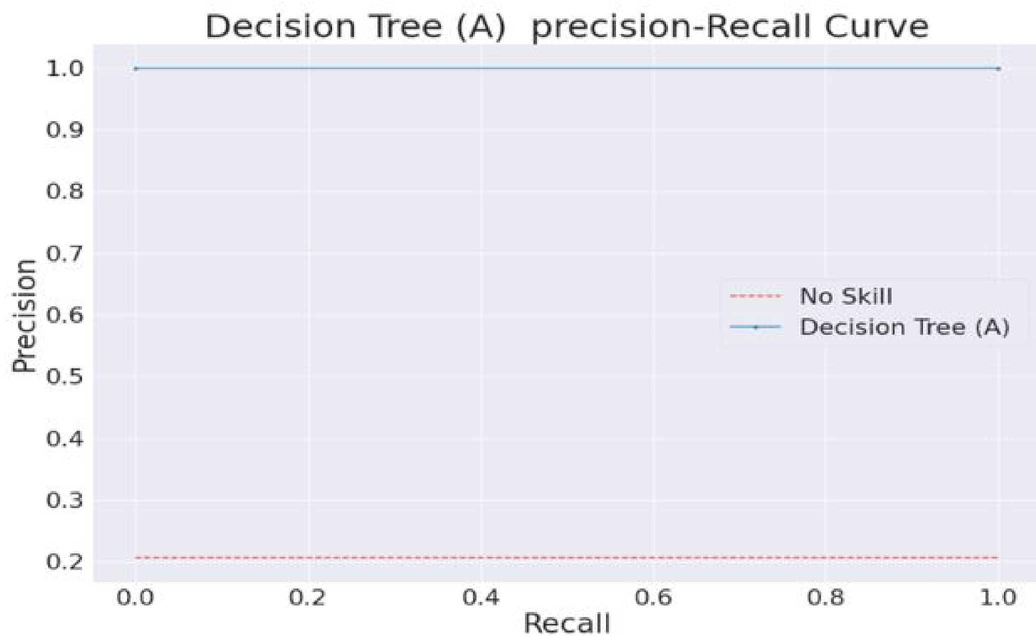

Fig. 4.3: Precision-Recall curve for decision tree classifier

4.2 Random Forest Classifier

Bellow is the summary of performance metrics for the Random Forest Classifier

Tab. 4.2: Summary of performance metrics for the Random Forest Classifier

Model Name	Machine Condition	Precision	Recall	F_1_Score	Accuracy	AUC
Random Forest Classifier	Failed (1)	1.0	1.0	1.0	100%	1.0
	Normal (0)	1.0	1.0	1.0		

The same training and test datasets were used to train and validate the performance of the random forest classifier. As shown in Table (4.2) the model had similar performance to the decision tree classifier on the unseen data as it recorded a 100% accuracy. Appendix B shows the ROC-AUC curve and the precision-recall curve for this model. It is worthy of note that this research was conducted using big data which tends to favour the performance of some models.

4.3 Support Vector Machine Classifier (SVM)

Bellow is the summary of performance metrics for Support Vector Machine classifier

Tab. 4.3: Summary of performance metrics for the Support Vector Machine classifier

Model Name	**Machine Condition**	**Precision**	**Recall**	F_1**_Score**	**Accuracy**	**AUC**
SVM Validation Set parameters tuned	Failed(1)	0.99	0.82	0.90	96%	
	Normal (0)	0.95	0.82	0.90		
SVM Test with tuned parameters	Failed (1)	0.99	0.83	0.90	0.96%	0.998
	Normal (0)	0.96	1.00	0.98		
SVM Validation No parameter tuning	Failed (1)	1.0	0.36	0.53	85%	
	Normal (0)	0.84	1.0	0.91		
SVM Test No parameter tuning	Failed (1)	1.0	0.40	0.57	88%	0.981
	Normal (0)	.087	1.0	0.93		

The support vector classifier when run without parameter tuning, an accuracy score of 88% was obtained on the test dataset. The recall for our desired class (machine failure) is only 0.4 for the test dataset. This value is close to the recall of 0.36 obtained for the validation set when the parameters are not yet tuned. This shows that the model does not overfit but generalizes well on unseen data. To improve the performance, we used GridSearchCV to perform parameter tuning for the model. For a limited range of values selected, the best SVM model obtained scored 96% accuracy on the unseen test data as reported in Table (4.3). The recall more doubled from a value of 0.40 for the test dataset without parameter tuning to a value of 0.83, with F_1 scores also witnessing an increase from 0.57 to 0.90. We achieved a generalization capability that is double the one obtained when a model with no parameter tuning is used as we were able to correctly predict 83% of machine failure cases up from 40% previously obtained.

Figure (4.4), Figure (4.5) and Figure (4.6) show the ROC-AUC curve, precision-recall curve and the confusion matrix respectively, for the support vector classifier for both cases: tuned parameters and parameters not tuned. From Figure (4.6), we see that the model without tuned parameters misclassified 3452 failure conditions as normal machine conditions. By performing parameter tuning, the error was reduced and only 973 misclassified cases obtained, achieving an improvement of over 71.8%.

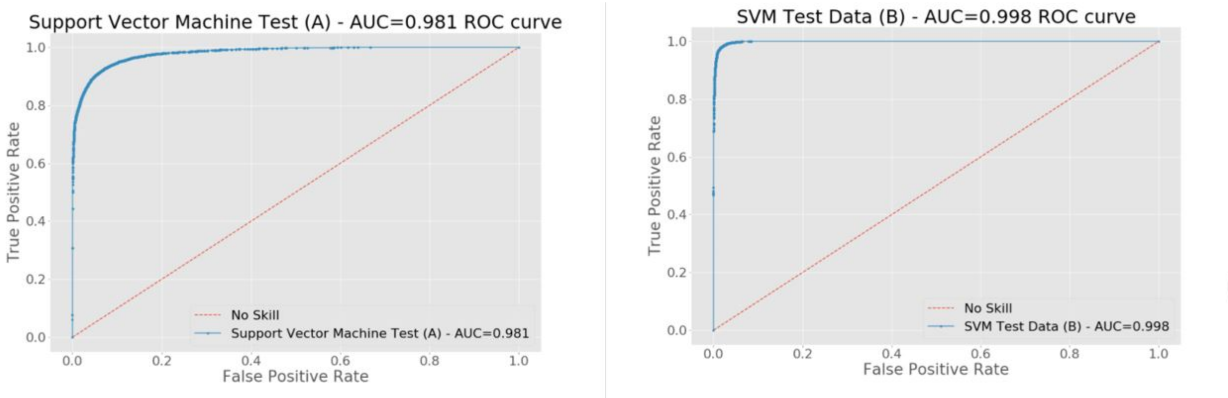

Fig. 4.4: ROC-AUC for Support Vector Machine Classifier

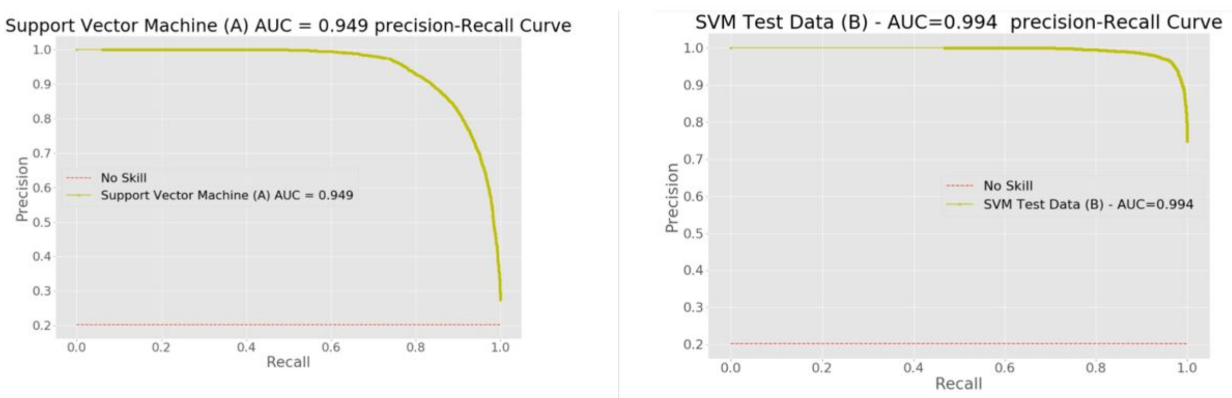

Fig. 4.5: Precision-Recall curve for Support Vector Machine Classifier

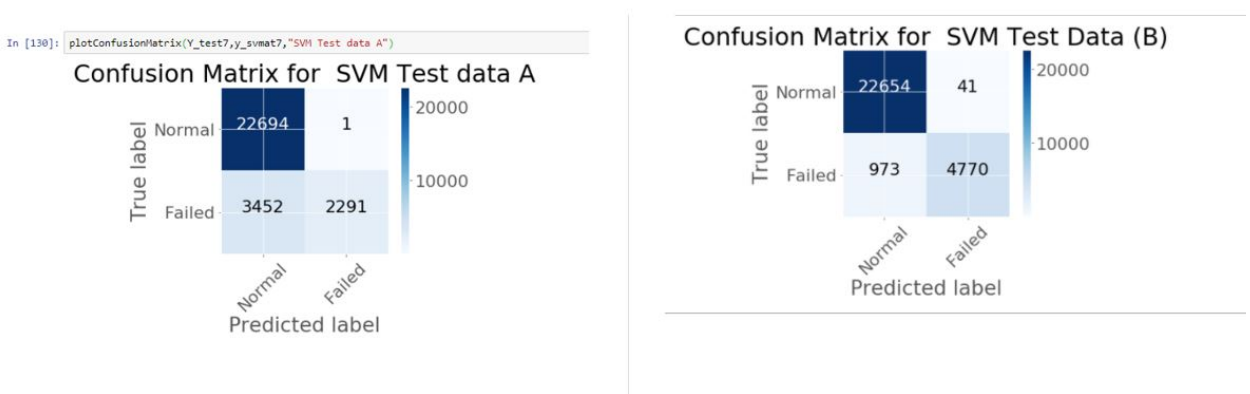

Fig. 4.6: Confusion matrix for Support Vector Machine Classifier

4.4 K-Nearest Neighbors (KNN)

Bellow is the summary of performance metrics for K-Nearest Neighbors classifier

Tab. 4.4: Summary of performance metrics for the K-Nearest Neighbors classifier

Model Name	**Machine Condition**	**Precision**	**Recall**	F_1**_Score**	**Accuracy**	**AUC**
KNN Validation with Tuned parameters	Failed (1)	0.91	0.62	0.74	90%	
	Normal (0)	0.01	0.98	0.94		
KNN Test with Tuned parameters	Failed (1)	0.90	0.65	0.75	91%	0.913
	Normal (0)	0.92	0.98	0.95		
KNN Validation No Tuning	Failed (1)	0.92	0.62	0.73	90%	
	Normal (0)	0.89	0.98	0.94		
KNN Test Data No parameter tuning	Failed (1)	0.90	0.65	0.75	91%	0.905
	Normal (0)	0.92	0.98	0.95		

By building a K-Nearest Neighbours classifier without considering tuning the parameters, we obtained a validation accuracy of 91% on the test dataset. The recall of this model was 0.61, meaning that the model could only correctly classify 65% of records that represent conditions during which a machine failure occurred. After conducting parameter tuning of the KNN, the resulting model had a similar accuracy score of 91% as for the model without parameter tuning. Also, the recall does not change. It is worthy of note that for the given data, the performance of the SVM is better than that of the KNN classifier. Figure (**??**) and Figure (**??**) show the ROC-AUC curves and the precision-recall curves respectively, for the KNN model.

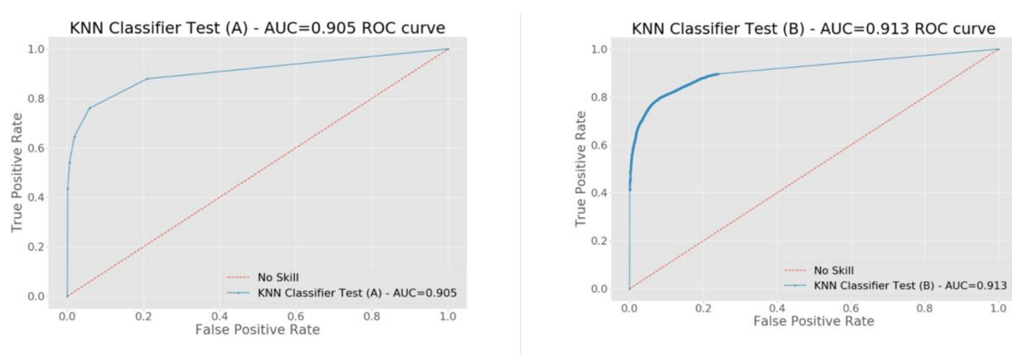

Fig. 4.7: ROC-AUC for Support Vector Machine Classifier

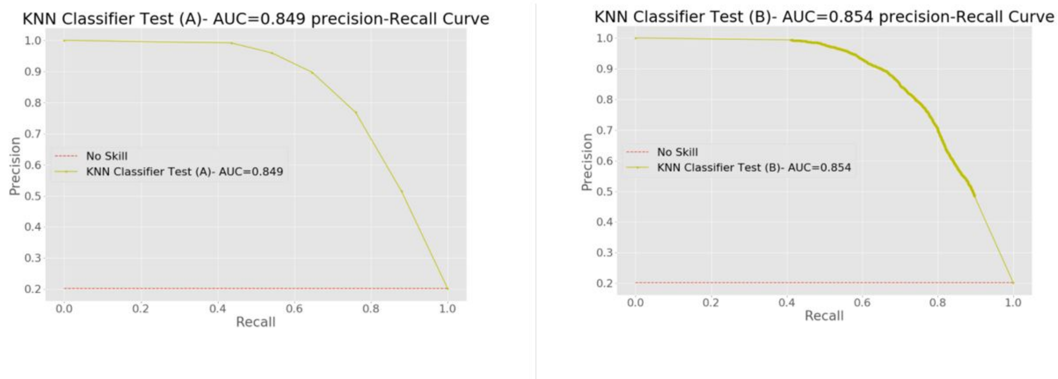

Fig. 4.8: Precision-Recall curve for Support Vector Machine Classifier

4.5 Artificial Neural Network: Multilayer Perceptron Classifier

Bellow is the summary of performance metrics for the Multilayer Perceptron Classifier

Tab. 4.5: Summary of performance metrics for the Multilayer perceptron Classifier

Model Name	**Machine Condition**	**Precision**	**Recall**	F_1 **Score**	**Accuracy**	**AUC**
MLP Classifier No Tuning Validation set	Failed (1)	0.70	0.47	0.56	83%	0.852
	Normal (0)	0.85	0.94	0.89		
MLP Classifier No Tuning Test set	Failed (1)	0.77	0.42	0.54	83%	0.873
	Normal (0)	0.87	0.97	0.92		
MLP Validation Data Tuned parameter	Failed (1)	1.0	0.97	0.99	99%	
	Normal (0)	0.99	1.0	1.0		
MLP Test Data Tuned parameter	Failed (1)	1.0	0.98	0.99	99.999...%	0.9999997
	Normal (0)	0.99	1.0	1.0		

The Multilayer Perceptron neural when trained and evaluated on the validation and test datasets without parameter tuning recorded 83% accuracy. The recall witnessed a slight drop from 47% for the validation set to 42% for the test set.

The parameters of the neural network were then tuned with the following values obtained as optimal: $activation = \text{'}tanh'$, $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $learning\_rate = 0.001$,$solver = \text{'}adam'$, $validation\_fraction = 0.1$, $batch\_size = \text{'}auto'$, $hidden\_layer_sizes = (50, 100, 50)$. When tested on the unseen data, the model scored an accuracy of 99.99994% on the test set, which is very similar to the 99% accuracy on the validation. The model performed extremely well after parameter tuning as the recall increased from 42% to 98% on the test set. The closely related accuracy values between the validation and the test datasets shows the model generalizes very well.

The ROC-AUC curve and the precision-recall and the best parameters of the model are shown in Appendix C.

4.6 Discussion

In Table (4.6) a summary of the models' performances is presented. Figure 4.5 shows the ROC-AUC curves for the MLP, KNN and SVM models plotted on the same pair of axes. Because of the behavior on the decision tree and the random forest models, we chose to not report the results but only mentioned them so that further investigation could be done. Of the three models considered, the Multilayer perceptron gave the best performance of 99.99994% accuracy when validated using the test data. The model we found with the least performance on the given dataset is the K-Nearest Neighbor Classifier. It recorded an accuracy score of 91% and recall of only 0.65. We note that due to limited compute power, attempts to improve model performance through parameter tuning by considering a wider range of values were unsuccessful. This might have impacted the performances of the SVM and KNN models positively. The recall scores of 0.65 and 0.83 for KNN classifier and SVM respectively may have improved if parameter tuning was successfully accomplished using a wider range of values as desired.

Susto et al. (2014) used a neural network to predict the failure/anomaly of direct current motors using the motion current signature and motor loads from generated data. These loads are grouped into five different classes and either classified as an anomaly or not. Their research yielded a 97.59% correct classification rate. Although we used a different dataset, we noticed that the best performed model in our case - Multilayer perceptron neural network scored a 99.999...% accuracy which is a better performance in our predicting system failure. Kolokas et al. (2018) used machine learning techniques to forecast fault occurrence in a production machine. Their best model could predict a fault 5 minutes to its occurrence at an accuracy less than 60%. In our case, we were able to predict machine failure at a greater horizon of 1 day. Our best model's accuracy is almost 100% and therefore generally performed better. Although we did not use the same datasets, it is clear that our research is significant as better accuracy scores are obtained.

Tab. 4.6: Summary of performance metrics for the three models

Model Name	Machine Condition	Precision	Recall	F_1_Score	Accuracy	AUC
SVM Test data tuned parameters	Failed (1)	0.99	0.83	0.90	96%	0.998
	Normal (0)	0.96	1.0	0.98		
K-NN Test Data Tuned parameters	Failed (1)	0.90	0.65	0.75	91%	0.913
	Normal (0)	0.92	0.98	0.95		
Multilayer Perceptron Classifier	Failed (1)	1.0	0.98	0.99	99.99994%	0.9999997
	Normal (0)	0.99	1.0	1.0		

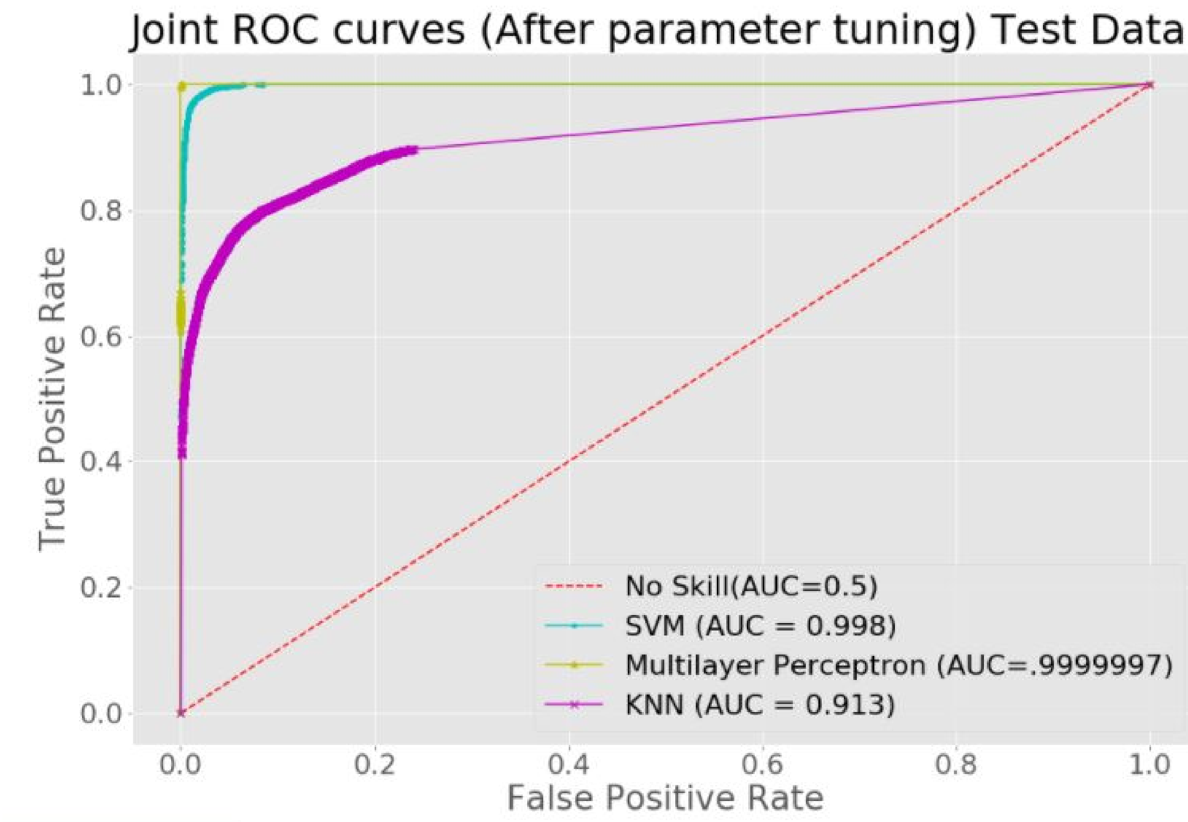

Fig. 4.9: ROC-AUC plots for all models used

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

We provided visibility into the actual wear and tear of manufacturing machines for improved maintenance practices that are less costly. In an attempt to mitigate interruption of the manufacturing process due to machine breakdowns and to keep systems running, we built machine learning algorithms and a neural network using historical machine failure data and used the algorithms to predict anomalies/system health issues one day before they occur. For the dataset used in this research, we were able to predict machine one day before it happens, providing useful insight to stakeholders for timely action. The multilayer perceptron classifier recorded an accuracy of almost 100% (99.99994%) when validated on the unseen test dataset.

The performances of our models were measured using various metrics including accuracy score, recall and area under the curve values. Because we are dealing with a classification problem where the data is imbalanced, the ideal metric to focus on is recall. The best performed model on the dataset scored 99.99994% for accuracy, 99.7% for recall, Area Under the Curve value of 0.9999997. By tuning parameters for the SVM, we were able to improve its recall by twice the original value from 0.40 to 0.83. This value is impressive as using this model, we can predict 83% of machine failures, reducing maintenance cost and disruption of operations considerably.

5.2 Recommendations and Future Works

This research was conducted using a publicly available dataset with no insight on how the data was captured or generated. As such, the effects of underlying unknown assumptions on the performance of the models used cannot be stated. Predictive maintenance model building being an uphill task, associating insight found in the dataset to real-life maintenance problems is a challenging process. Domain knowledge from experts that could guide the implementation of the models used in this research was missing. Future works could use such insight during data processing and that may improve the performance of the models further.

Another limitation of the research process is that we were unable to get expert knowledge that could guide us to provide insight as to which system part led to failure. This insight is very helpful as it facilitates the diagnosis and repair process. We recommend future work that could provide insight on the possible part that failed. This would further reduce the time spent during fault diagnosis and maintenance.

As stated earlier in chapter four, limited compute power restricted the range of values that were considered when conducting parameter tuning for the SVC and KNN classifiers. A research that considers an expanded set of values for the parameters could identify possible parameter combinations that would greatly improve the performances of these two models.

Finally, in this study, we have not leveraged the strength of cascaded multiple machine learning models to predict system maintenance. Literature suggests that cascading machine learning algorithms yields better performance when compared to the use of one method. By leveraging two or more techniques cascaded together, we could further reduce false positive rate for those models which did not perform at 100% accuracy.

Aussel, N., Jaulin, S., Gandon, G., Petetin, Y., Fazli, E., and Chabridon, S. (2017). Predictive models of hard drive failures based on operational data. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 619–625. IEEE.

Awad, M. and Khanna, R. (2015). Support vector machines for classification. In *Efficient Learning Machines*, pages 39–66. Springer.

Bashir, M., Awan, I. U., Ugail, H., and Muhammad, Y. (2019). Failure prediction using machine learning in a virtualised hpc system and application.

Bigml (Ap. 10, 2020). Machine Failure
https://bigml.com/dashboard/dataset/5ecbe369cb4f962b4c00261d.

Borgi, T., Hidri, A., Neef, B., and Naceur, M. S. (2017). Data analytics for predictive maintenance of industrial robots. In *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, pages 412–417. IEEE.

Choudhury, A. (May 27, 2020). 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset
https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/.

Christiansen, B. (Ap. 11, 2020). A Complete Guide To Predictive Maintenance
https://limblecmms.com/blog/predictive-maintenance/.

Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., and Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, 88(11):2783–2792.

Elliott, J., Kharbas, K., Fiala, D., Mueller, F., Ferreira, K., and Engelmann, C. (2012). Combining partial redundancy and checkpointing for hpc. In *2012 IEEE 32nd International Conference on Distributed Computing Systems*, pages 615–626. IEEE.

Gardner, M. W. and Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636.

Javatpoint (August 08, 2020). Decision Tree Classification Algorith
`https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm`.

Jordan, J. (May 27, 2020). Important Techniques to process imbalanced data in machine learning
`https://analyticsindiamag.com/5-important-techniques-to-process-imbalanced-data-in-machine-learning`.

Kaggle (2020). Machine Failure Data
`https://www.kaggle.com/binaicrai/machine-failure-data`.

Kanawaday, A. and Sane, A. (2017). Machine learning for predictive maintenance of industrial machines using iot sensor data. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 87–90. IEEE.

Kolokas, N., Vafeiadis, T., Ioannidis, D., and Tzovaras, D. (2018). Forecasting faults of industrial equipment using machine learning classifiers. In *2018 Innovations in Intelligent Systems and Applications (INISTA)*, pages 1–6. IEEE.

Langone, R., Alzate, C., Bey-Temsamani, A., and Suykens, J. A. (2014). Alarm prediction in industrial machines using autoregressive ls-svm models. In *2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 359–364. IEEE.

Luigi (May 26, 2020). How Data Leakage Impacts Machine Learning Models
`https://mlinproduction.com/data-leakage/`.

Luo, M., Xu, Z., Chan, H. L., and Alavi, M. (2013). Online predictive maintenance approach for semiconductor equipment. In *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*, pages 3662–3667. IEEE.

Martinez, J., Christianne, D., and Zhengyi, L. (May, 2020). Failure Prediction using Machine Learning in a Virtualised HPC System and application
`https://pdfs.semanticscholar.org/e91a/ea5830c61cfcfb12764871919d33595a510c.pdf`.

RENOVETEC (Ap. 11, 2020). What is predictive maintenance?
`http://www.mantenimientopetroquimica.com/en/rcm/144-articles-of-interest/100-what-is-predictive-maintenance.html`.

Robinson, G. M. (2009). *Time series Analysis, in International encyclopedia of human geography*. Elsevier.

Sennaar, K. (Ap. 05, 2020). Telecom Machine Learning Applications–Comparing AT&T, Verizon, Comcast and More
`https://emerj.com/ai-sector-overviews/telecom-machine-learning-applications`.

Simidu, U. (2001). An introduction to predictive microbiology. *Shokuhin eiseigaku zasshi. Journal of the Food Hygienic Society of Japan*, 42(6):J317.

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., and Beghi, A. (2014). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820.

Traini, E., Bruno, G., D'Antonio, G., and Lombardi, F. (2019). Machine learning framework for predictive maintenance in milling. *IFAC-PapersOnLine*, 52(13):177–182.

Tran, V. P. M. (May 25, 2020). Manufacturing Downtime Reduction with Predictive maintenance
`https://arimo.com/machine-learning/2016/manufacturing-downtime-cost -reduction-predictive-maintenance/`.

Yang, C., Letourneau, S., Liu, J., Cheng, Q., and Yang, Y. (2017). Machine learning-based methods for ttf estimation with application to apu prognostics. *Applied Intelligence*, 46(1):227–239.

Yang, K. and Shahabi, C. (2004). A pca-based similarity measure for multivariate time series. In *Proceedings of the 2nd ACM international workshop on Multimedia databases*, pages 65–74.

Yu, Y., Zhu, Y., Li, S., and Wan, D. (2014). Time series outlier detection based on sliding window prediction. *Mathematical problems in Engineering*, 2014.

APPENDICES

Appendix A : Random Forest classifier plots

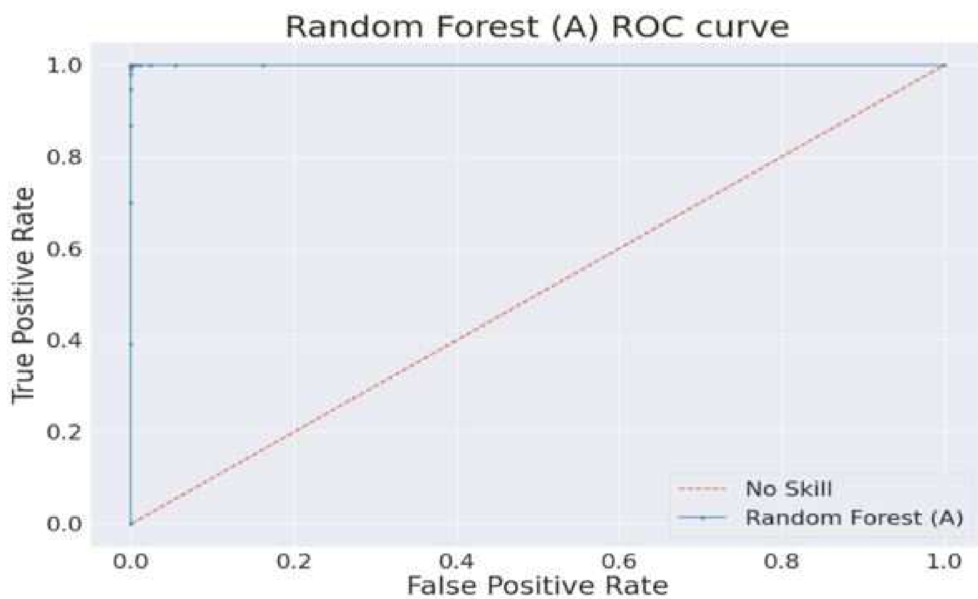

Fig. 5.1: RF classifier

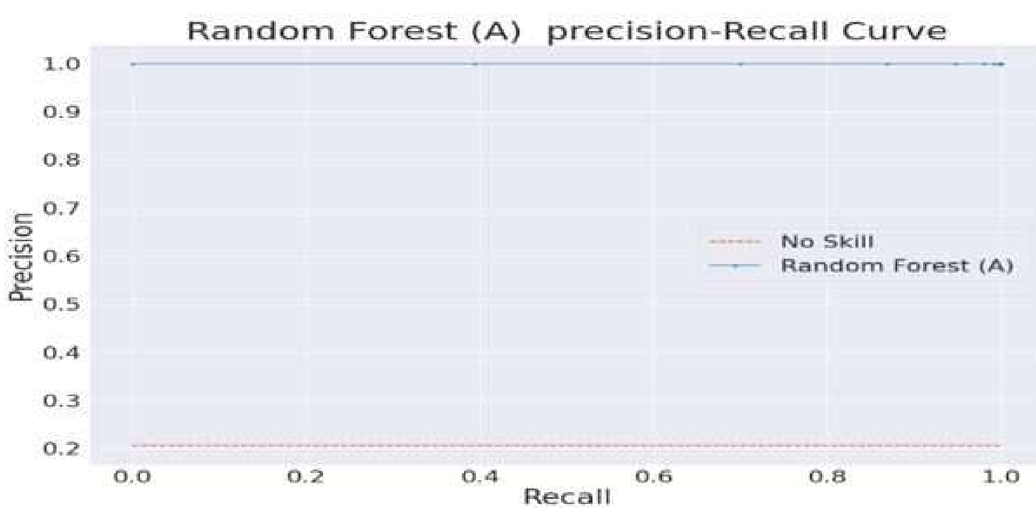

Fig. 5.2: RF classifier

Appendix B : Multilayer perceptron plots and parameters

```
Best Parameters:
 MLPClassifier(activation='tanh', alpha=0.0001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(50, 100, 50), learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=100,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=123, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

Fig. 5.3: Best parameter settings for the multilayer perceptron model

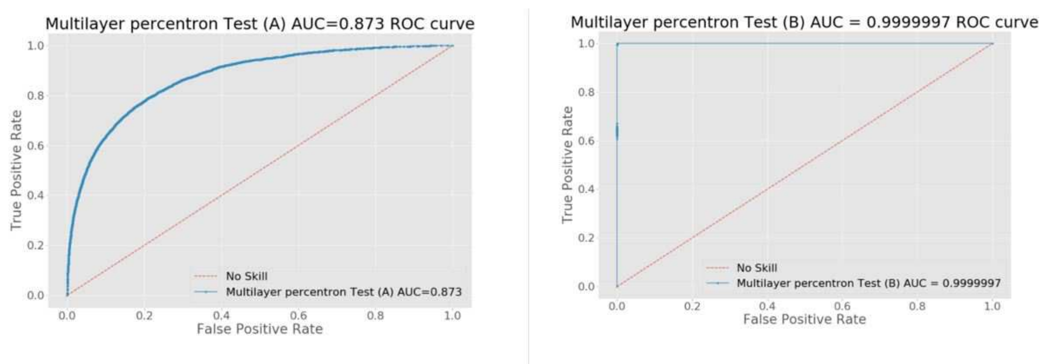

Fig. 5.4: ROC-AUC curve for Multilayer perceptron

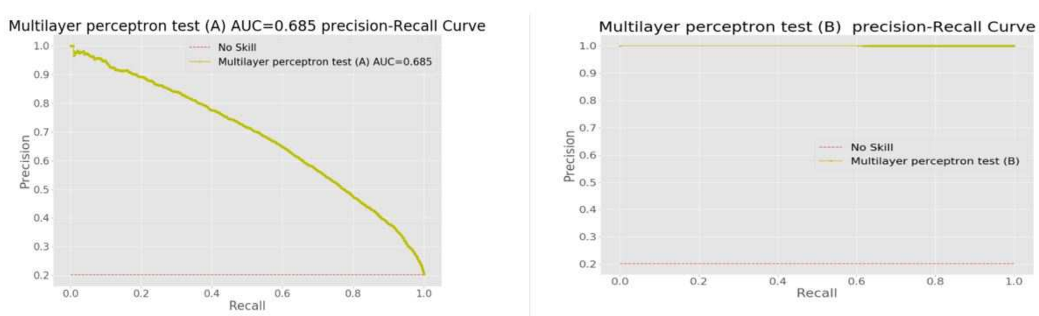

Fig. 5.5: Precision-recall curve for the Multilayer perceptron classifier

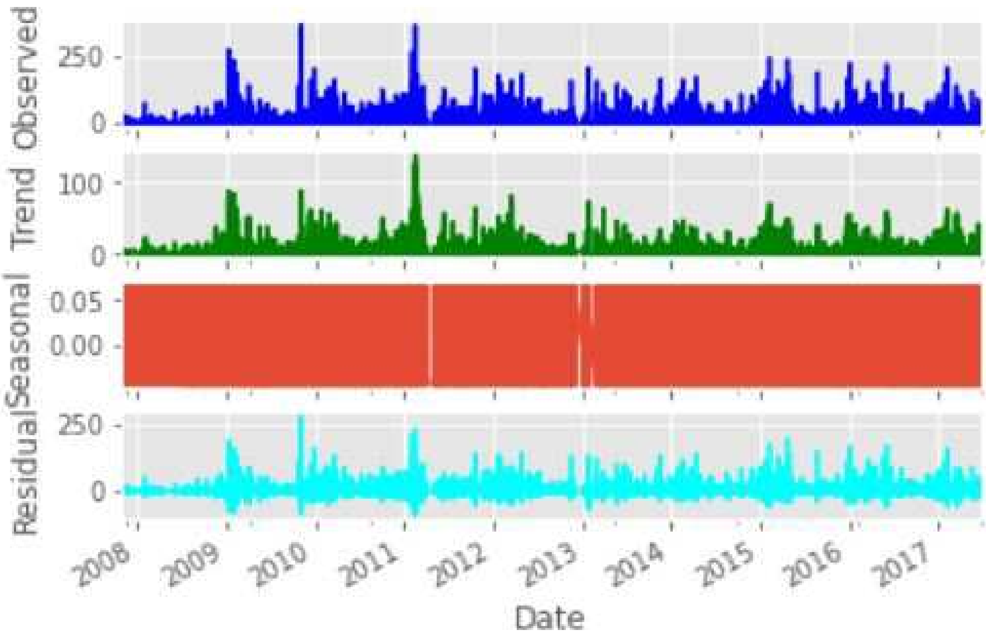

Fig. 5.6: Seasonal decomposition of data attribute $Risk\_MM$

We see that the data is noisy as shown by the two attributes after seasonal decomposition.

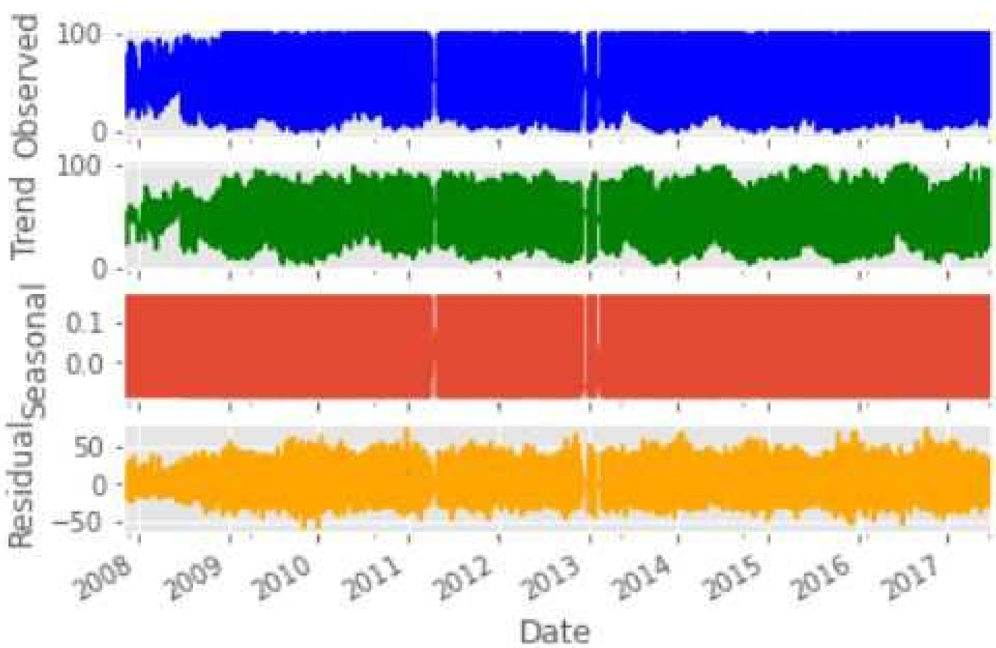

Fig. 5.7: Seasonal decomposition of data $Parameter4\_3pm$

Appendix C : Pair plot for selected variables

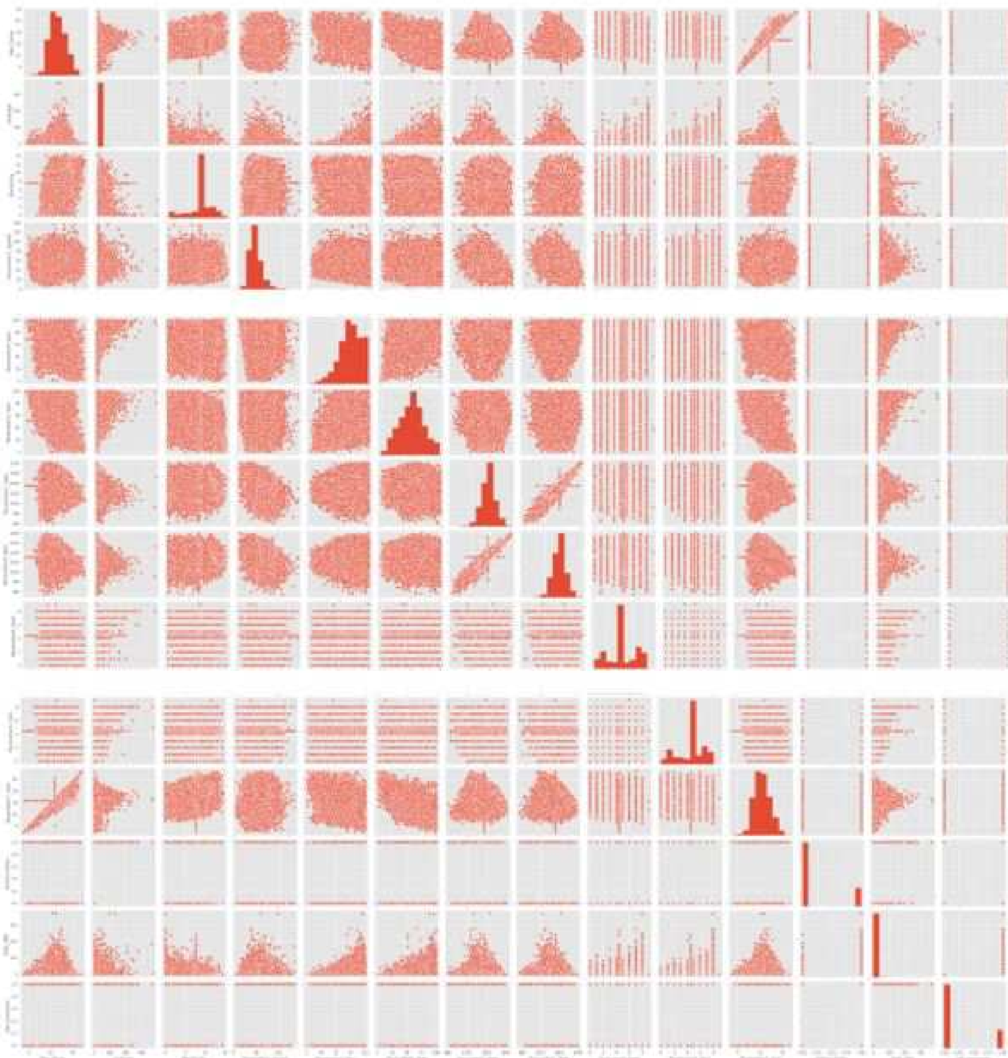

Fig. 5.8: Correlation coefficient matrix for the data independent variables

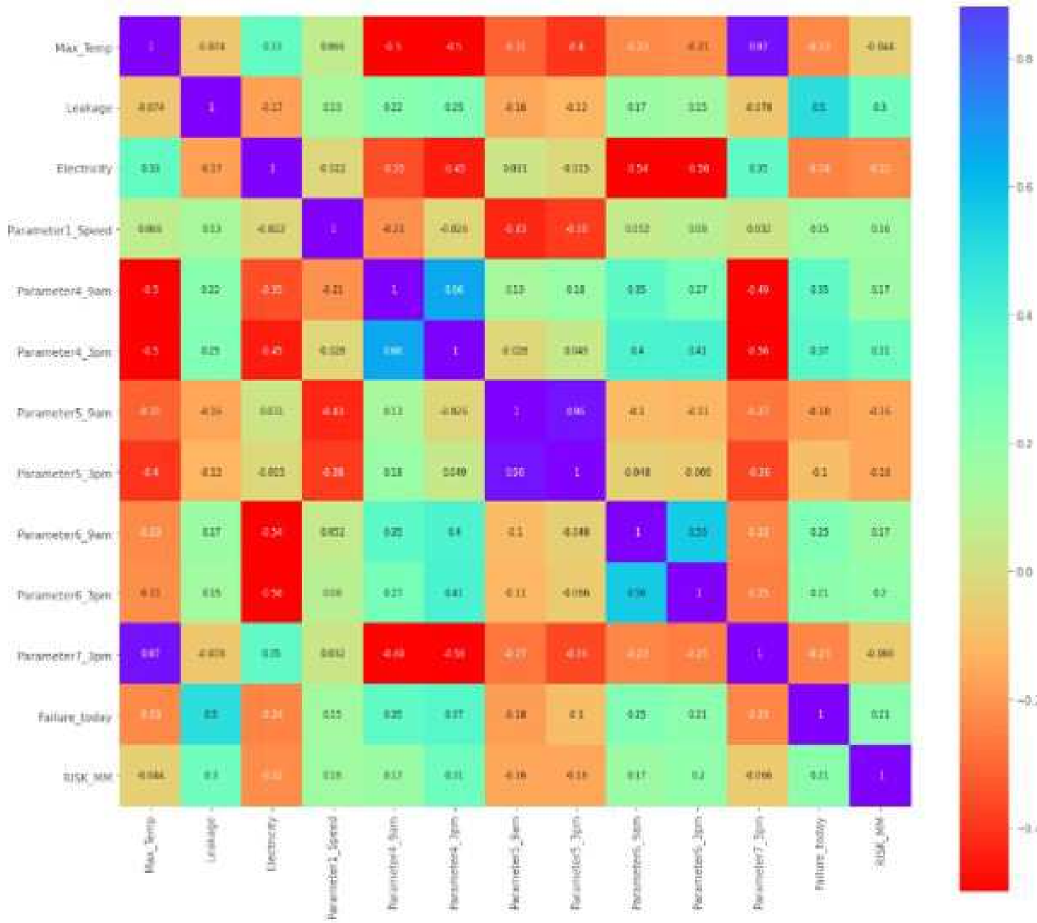

Fig. 5.9: Correlation coefficient matrix for the data independent variables

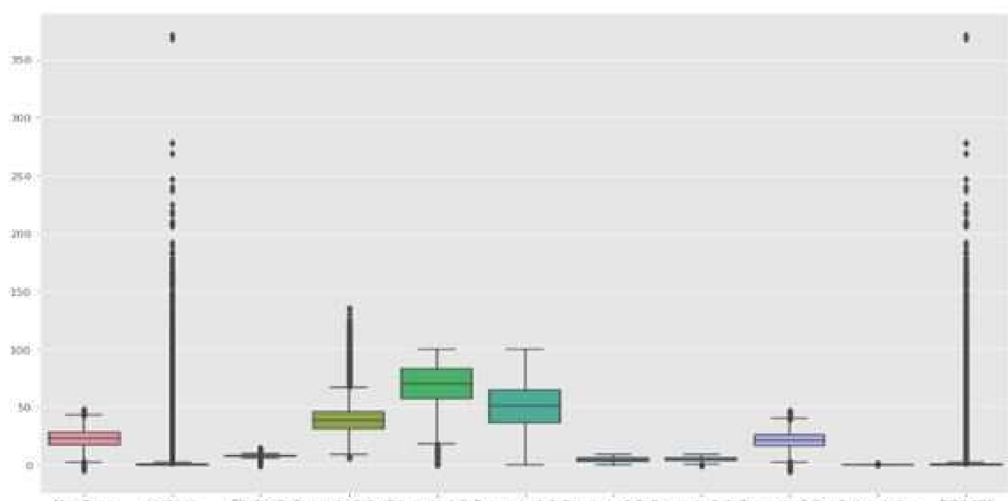

Fig. 5.10: Boxplot of some independent variables

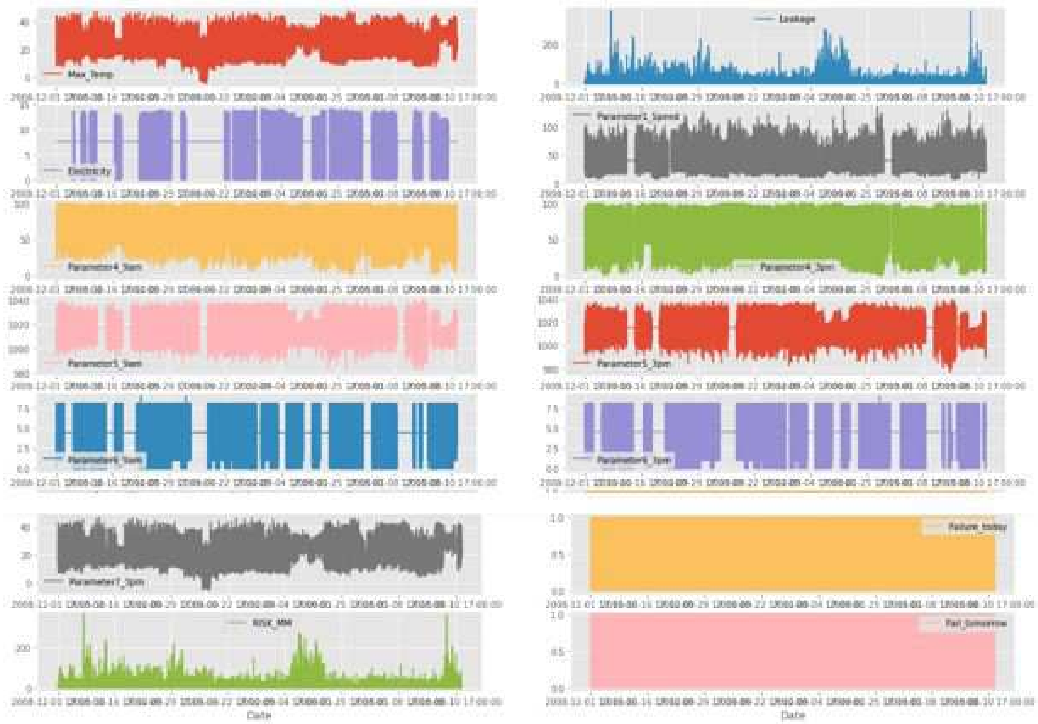

Fig. 5.11: Visualizing the variation of recorded data for each feature over time

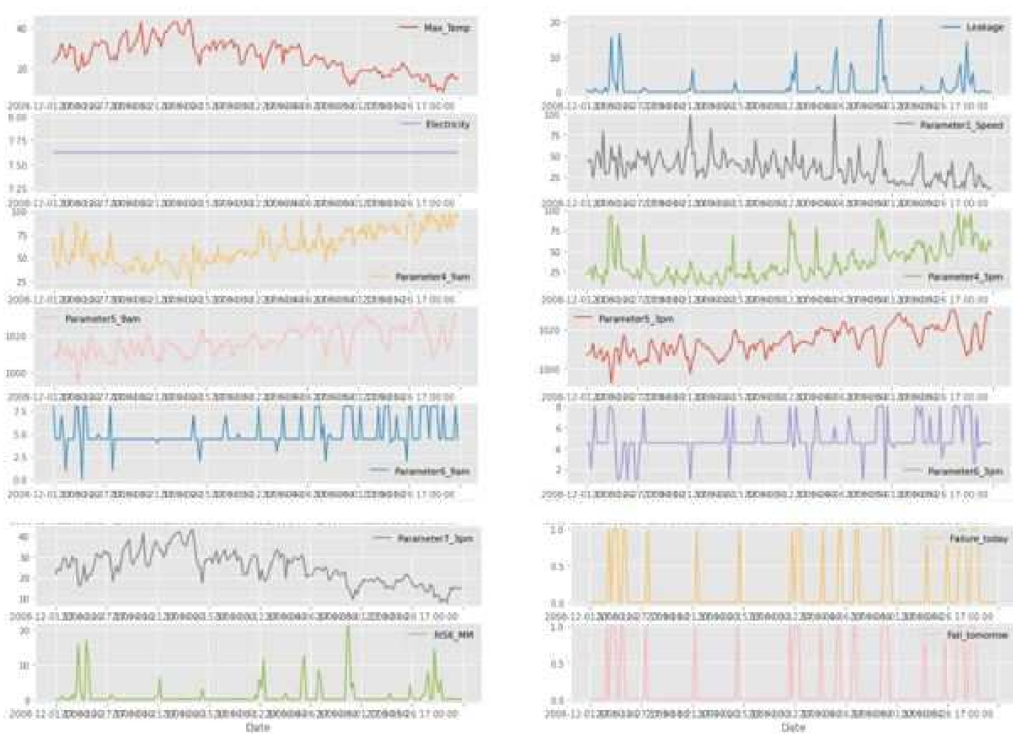

Fig. 5.12: Zoomed in visualization of data feature variation with time

NGWA_PIUS_MS_Thesis

ORIGINALITY REPORT

22%	17%	16%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	link.springer.com Internet Source	2%
2	www.hindawi.com Internet Source	1%
3	Adele Cutler. "Random Forests", Ensemble Machine Learning, 2012 Publication	1%
4	nparc.cisti-icist.nrc-cnrc.gc.ca Internet Source	1%
5	pureadmin.qub.ac.uk Internet Source	1%
6	Submitted to Graphic Era University Student Paper	<1%
7	limblecmms.com Internet Source	<1%
8	machinelearningmastery.com Internet Source	<1%
9	Submitted to Swinburne University of	

Technology
Student Paper
<1%

10 Submitted to University of Wales Institute, Cardiff
Student Paper
<1%

11 Tawfik Borgi, Adel Hidri, Benjamin Neef, Mohamed Saber Naceur. "Data analytics for predictive maintenance of industrial robots", 2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET), 2017
Publication
<1%

12 Submitted to Liverpool John Moores University
Student Paper
<1%

13 Willamos Silva, Miriam Capretz. "Assets Predictive Maintenance Using Convolutional Neural Networks", 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2019
Publication
<1%

14 "Advances in Artificial Intelligence and Data Engineering", Springer Science and Business Media LLC, 2021
Publication
<1%

15	repozitorij.foi.unizg.hr Internet Source	<1%
16	Submitted to University of Westminster Student Paper	<1%
17	Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Sean McLoone, Alessandro Beghi. "Machine Learning for Predictive Maintenance: A Multiple Classifier Approach", IEEE Transactions on Industrial Informatics, 2015 Publication	<1%
18	export.arxiv.org Internet Source	<1%
19	Submitted to Eastern Institute of Technology Student Paper	<1%
20	Submitted to University of Surrey Student Paper	<1%
21	Kevin Villalobos, Johan Suykens, Arantza Illarramendi. "A flexible alarm prediction system for smart manufacturing scenarios following a forecaster–analyzer approach", Journal of Intelligent Manufacturing, 2020 Publication	<1%
22	www.javatpoint.com Internet Source	<1%
23	ir.uitm.edu.my	

Internet Source <1%

24	Submitted to University of West Florida	Student Paper	<1%
25	Submitted to Cardiff University	Student Paper	<1%
26	Submitted to South Cheshire College	Student Paper	<1%
27	tel.archives-ouvertes.fr	Internet Source	<1%
28	edoc.pub	Internet Source	<1%
29	hal.archives-ouvertes.fr	Internet Source	<1%
30	Submitted to University of Wollongong	Student Paper	<1%
31	www.lib.ncsu.edu	Internet Source	<1%
32	Rocco Langone, Carlos Alzate, Abdellatif Bey-Temsamani, Johan A. K. Suykens. "Alarm prediction in industrial machines using autoregressive LS-SVM models", 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 2014	Publication	<1%

33	Submitted to Kingston University Student Paper	<1%
34	"International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2021 Publication	<1%
35	Submitted to Leeds Metropolitan University Student Paper	<1%
36	Emiliano Traini, Giulia Bruno, Gianluca D’Antonio, Franco Lombardi. "Machine Learning Framework for Predictive Maintenance in Milling", IFAC-PapersOnLine, 2019 Publication	<1%
37	gc.sfc.keio.ac.jp Internet Source	<1%
38	Submitted to University College London Student Paper	<1%
39	Submitted to Hellenic Open University Student Paper	<1%
40	mlinproduction.com Internet Source	<1%
41	www.wjgnet.com Internet Source	<1%
42	Submitted to Pearson College Student Paper	

<1%

43	Submitted to University of Edinburgh Student Paper	<1%
44	Dipanjan Sarkar, Raghav Bali, Tushar Sharma. "Practical Machine Learning with Python", Springer Science and Business Media LLC, 2018 Publication	<1%
45	pdfs.semanticscholar.org Internet Source	<1%
46	www.readkong.com Internet Source	<1%
47	Submitted to University of Leicester Student Paper	<1%
48	Submitted to University of Sunderland Student Paper	<1%
49	Submitted to Queen Mary and Westfield College Student Paper	<1%
50	Afreen Khan, Swaleha Zubair. "An Improved Multi-Modal based Machine Learning Approach for the Prognosis of Alzheimer's disease", Journal of King Saud University - Computer and Information Sciences, 2020 Publication	<1%

51	analyticsindiamag.com Internet Source	<1%
52	docplayer.net Internet Source	<1%
53	www.freepatentsonline.com Internet Source	<1%
54	Lovedeep Ramana, Wooram Choi, Young-Jin Cha. "Chapter 4 Automated Vision-Based Loosened Bolt Detection Using the Cascade Detector", Springer Science and Business Media LLC, 2017 Publication	<1%
55	Submitted to University of Glasgow Student Paper	<1%
56	eprints.usm.my Internet Source	<1%
57	www.scribd.com Internet Source	<1%
58	ideas.repec.org Internet Source	<1%
59	Submitted to Clemson University Student Paper	<1%
60	espace.library.uq.edu.au Internet Source	<1%

61	Submitted to Central Queensland University Student Paper	<1%
62	www.slideshare.net Internet Source	<1%
63	Submitted to Universitat Politècnica de València Student Paper	<1%
64	koara.lib.keio.ac.jp Internet Source	<1%
65	www.inderscienceonline.com Internet Source	<1%
66	Submitted to Ashesi University Student Paper	<1%
67	trap.ncirl.ie Internet Source	<1%
68	nadre.ethernet.edu.et Internet Source	<1%
69	Submitted to The Hong Kong Polytechnic University Student Paper	<1%
70	Submitted to Coventry University Student Paper	<1%
71	Submitted to Institute of Technology Carlow Student Paper	<1%

72	"Computer Information Systems and Industrial Management", Springer Science and Business Media LLC, 2020 Publication	<1%
73	univagora.ro Internet Source	<1%
74	www.cluteinstitute.com Internet Source	<1%
75	Submitted to University of Technology, Sydney Student Paper	<1%
76	Submitted to University of Huddersfield Student Paper	<1%
77	repository.tudelft.nl Internet Source	<1%
78	Mohd Aijaj Khan, Manish Dixit, Aaradhya Dixit. "Demystifying and Anticipating Graduate School Admissions using Machine Learning Algorithms", 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), 2020 Publication	<1%
79	Nicole Nichols, Mark Greaves, William Smith, Ryan LaMothe, Gianluca Longoni, Jeremy Teuton. "Identification of program signatures from cloud computing system telemetry data",	<1%

2016 IEEE Symposium Series on Computational Intelligence (SSCI), 2016
Publication

80	Santanu Pattanayak. "Pro Deep Learning with TensorFlow", Springer Science and Business Media LLC, 2017 Publication	<1 %
81	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1 %
82	Vanyashree Mardi, Naresh E., Vijaya Kumar B. P.. "chapter 8 A Survey on Deep Learning Techniques Used for Quality Process", IGI Global, 2019 Publication	<1 %
83	Submitted to Suan Dusit Rajabhat University Student Paper	<1 %
84	Jose Cleydson F. Silva, Ruan M. Teixeira, Fabyano F. Silva, Sergio H. Brommonschenkel, Elizabeth P.B. Fontes. "Machine learning approaches and their current application in plant molecular biology: A systematic review", Plant Science, 2019 Publication	<1 %
85	Xilei Zhao, Ruggiero Lovreglio, Daniel Nilsson. "Modelling and interpreting pre-evacuation decision-making using machine learning",	<1 %

Automation in Construction, 2020
Publication

86 phd.lib.uni-corvinus.hu
Internet Source
<1%

87 Ava Kwong. "Accuracy of BRCA1/2 Mutation Prediction Models for Different Ethnicities and Genders: Experience in a Southern Chinese Cohort", World Journal of Surgery, 01/31/2012
Publication
<1%

88 Submitted to University of Teesside
Student Paper
<1%

89 es.scribd.com
Internet Source
<1%

90 N. Saranya, V. Manikandan. "On viability of detecting malwares online using ensemble classification method with performance metrics", Computational Intelligence, 2020
Publication
<1%

91 Huihua Lu, Bojan Cukic, Mark Culp. "A Semi-supervised Approach to Software Defect Prediction", 2014 IEEE 38th Annual Computer Software and Applications Conference, 2014
Publication
<1%

92 psasir.upm.edu.my
Internet Source
<1%

www.datavedas.com

93	Internet Source	<1%
94	Jasmina Novakovic, Suzana Markovic. "Performance of Support Vector Machine in Imbalanced Data Set", 2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH), 2020 Publication	<1%
95	"Intelligent Computing, Networking, and Informatics", Springer Science and Business Media LLC, 2014 Publication	<1%
96	www.research-collection.ethz.ch Internet Source	<1%
97	www.nejm.org Internet Source	<1%
98	eprints.ums.edu.my Internet Source	<1%
99	scholarworks.sjsu.edu Internet Source	<1%
100	s-space.snu.ac.kr Internet Source	<1%
101	Muhammad Adnan, Asad Habib, Jawad Ashraf, Babar Shah, Gohar Ali. "Improving M-learners' Performance Through Deep Learning	<1%

Techniques by Leveraging Features Weights", IEEE Access, 2020
Publication

102 www.researchsquare.com
Internet Source
<1%

103 Pelagatti, Matteo. "Time Series Concepts", Time Series Modelling with Unobserved Components, 2015.
Publication
<1%

104 dspace.ucuenca.edu.ec
Internet Source
<1%

105 "Advanced Computing Technologies and Applications", Springer Science and Business Media LLC, 2020
Publication
<1%

106 James Connan. "KernTune", Proceedings of the 2007 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries - SAICSIT 07 SAICSIT 07, 2007
Publication
<1%

107 "Euro-Par 2014: Parallel Processing Workshops", Springer Science and Business Media LLC, 2014
Publication
<1%