People's **D**emocratic **R**epublic of **A**lgeria
Ministry of **H**igher **E**ducation and **S**cientific **R**esearch
Tlemcen university
Faculty of **T**echnology

Departement of Electrical and Electronics Engineering
Speciality Automation
Option: Industrial Computer Science

Master's Thesis

Predictive Maintenance Using Machine Learning

Presented by:

Mr INNOCENT MATEYAUNGA

Defended on 27/09/2020 infront of the jury

PRESIDENT :	Mr A. Hadj Abdelkader	PR, Tlemcen university
EXAMINATOR :	Mr H. Kahouadji	MCB, Tlemcen university
SUPERVISOR :	Mrs W. Handouzi	MCB, Tlemcen university

2019/2020

Acknowlegements

First and foremost,praises and thanks to God, the Almighty, for his guidance throughout my research enabling me to complete successfully.

Secondly, i would like to express my deepest and sincere gratitude to my supervisor **Dr. W.Handouzi** for providing me with invaluable guidance throughout my research. She was patient enough to take me through all procedures that facilitated me to complete this research. It was a great pleasure and honour to work under her guidance. She not only taught me the methodology to use but went a step further to motivate me and to offer friendship and empathy.

I would also like to express my honor and gratitude to the members of the jury, **Pr A. Hadj-Abdelkader** and **Dr H. Kahouadji** for having accepted to examine and evaluate my work.

Finally am extremely grateful to my friends and family for the continuous prayers, love, care and sacrifices made, pi wouldn't have done it without you guys.

Abstract

In the recent years Deep Learning has gained increasing attention, hence it's application in predictive maintenance. It is an evolving research area with a potential to influence diverse application domains and hence its use for system health management applications must put to research. This thesis presents a systematic review of Artificial Intelligence (AI) based system health management with emphasis on recent trends of Deep Learning (DL) within the field : different kinds of Machine Learning (ML) models which can be used to predict failure of a machine.At first a review of previous publications is carried out, then different models (LSTM,GRU and SRN) are employed on the NASA Turbofan data-set so as to determine the best model, CuDNN (CuDNNLSTM, CuDNNGRU) versions are also applied and time of execution is compared with the basic ones (LSTM, GRU). The CuDNNGRU is found to be the best method due to it's time of execution (30.6s) and it's competitive accuracy (0.9827) and precision (0.9452).

Key words : machine learning, artificial intelligence, failure , LSTM, GRU, SRN, predictive maintenance, deep learning.

Résumé

Ces dernières années, le Deep Learning a attiré une attention croissante, d'où son application dans la maintenance prédictive. Il s'agit d'un domaine de recherche en évolution qui a le potentiel d'influencer divers domaines d'application et, par conséquent, son utilisation pour les applications de gestion de la santé des systèmes et équipements complexes et couteux doit être mis à profit. Cette thèse présente une revue systématique de la gestion de la santé du système basée sur l'intelligence artificielle (IA) en mettant l'accent sur les tendances récentes de l'apprentissage profond (DL) dans le domaine: différents types de modèles d'apprentissage automatique (ML) qui

peuvent être utilisés pour prédire la défaillance d'une machine .Au début, la revue des publications précédentes est effectué, puis différents modèles (LSTM, GRU et SRN) sont utilisés sur l'ensemble de données NASA Turbofan afin de déterminer le meilleur modèle, les versions CuDNN (CuDNNLSTM, CuDNNGRU) sont également appliquées et le temps d'exécution est comparé aux temps de base (LSTM, GRU). Le CuDNNGRU s'avère être la meilleure méthode en raison de son temps d'exécution (30,6 s) et de sa exactitude compétitive (0,9827) et de sa précision (0,9452).

Mot clés : Apprentissage automatique, L'intelligence artificielle, defaillances, LSTM, GRU, SRN, maintenance prédictive, L'apprentissage en profondeur.

Contents

List of Tables

List of Figures

Nomenclature

AI Artificial Technology

C-MAPSS Commercial Modular Aero Propulsion System Simulation

CBLSTM Convolutional Bi-Directional LSTM

CM Corrective Maintenance

CNN Convolutional Neuron Networks

CRM Customer Relationship Management

DL Deep Learning

DNN Deep Neural Network

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

IoT Internet Of Things

LR Linear Regression

LSTM Long Short Term Memory

MLP Multi Layer Perceptron

MODBNE Multi-objective Deep Belief Networks Ensemble

PM Preventive Maintenance

RM Reactive Maintenance

RNN Recurrent Neuron Network

RUL Remaining Useful Life

SVM Support Vector Machines

SVR Support Vector Regression

TPU Tensor Processing Unit

Chapter 1

Theoretical Framework

Introduction

In this chapter we are going to look at the reasons why machines ought to be maintained and the types of maintenance that can be carried out.We are going to see how the evolution of modern techniques (e.g., Internet of things,sensing technology, artificial intelligence, etc.) influenced a transition of maintenance strategies from Reactive Maintenance (RM) to Corrective Maintenance(CM) to Predetermined Maintenance to Condition –based maintenance to Preventive Maintenance (PM) and finally to Predictive Maintenance (PdM). RM is only executed to restore the operating state of the equipment after failure occurs, and thus tends to cause serious lag and results in high reactive repair costs.PM is carried out according to a planned schedule based on time or process iterations to prevent breakdown, and thus may perform unnecessary maintenance and result in high prevention costs.In order to achieve the best trade-off between the two, PdM is performed based on an online estimate of the "health" and can achieve timely pre-failure interventions.In the same chapter we are also going to look at machine learning and Deep Learning.

1.1 Maintenance

Maintenance, can be generally defined as the efforts or actions taken either to maintain or improve the condition and performance of a machine.To fully

understand the different stages of maintenance, we need to define a few concepts :

- Symptom: it is the abnormal deviation of an observable quantity
- Fault: it is the abnormal deviation of at least one characteristics of a component or a system.
- Failure: it is when a component or a system stops permanently to perform its function.Partial failure: deterioration of the ability of a system to perform the required function.
- Complete failure: cessation of the ability of a system to perform the required function [12].

The main goal of health management is relevant data collection from sensors, processing the data which includes key feature extraction, fault diagnosis and prediction. Based on this analysis, the system will be able to recommend further actions according to user requirements. Recommended actions will be issued: fault alarms, the human operator evaluates the risk associated with the fault detected and will either choose to delay any action – if the failure can be tolerated until the next scheduled maintenance, or take an immediate action e.g. in the case of failures that can affect safety.This process has been illustrated in Fig **??**. A challenge often encountered in the implementation of algorithms is normally as a result of uncertainties in the raw data collected by the sensors. False alarms are a major annoyance during maintenance activities [13]. This is when there is a fault calls, when no actual fault exists, or a call for a maintenance action when none is needed. System models and related algorithms that are used for health management need to be updated from time-to-time so as to be able to account for any unanticipated conditions. Typically, once a system model is developed it would remain unchanged, therefore it is important to adapt models and algorithms to the

in-service performance, recording and storing acquired on-field knowledge so it can be used for future application developments and improvements. There are often problems associated with collection of meaningful information and analysis of all the acquired knowledge to improve diagnosis and resilience, these can render acquired data unusable, include: missing attributes, where several parameters may not have been measured during failure manifestation, improper data format, corrupt data, bad sensors or even the human operator errors.

The essential components for the identification of abnormalities and defective conditions include condition monitoring, detection, diagnosis and prognosis. Condition monitoring consists of monitoring the parameters to detect anomalies, while the detection mechanism would make it possible to detect these anomalies. The diagnostic process will be performed to determine where this abnormality is located, so that the necessary action can be taken. Finally, prognosis is a process of predicting and estimating the RUL of the system based on its performance. Therefore, it is common practice to define some minimum specifications for the operating health management system. This may include details about the operating environment, sensor tolerance, confidence levels, etc.

1.1.1 Why Should Machines Be maintained?

a. To prevent faults and breakdown

A machine should be regularly serviced, this significantly reduces the effects of faulty machines as it ensures that parts are still intact and use-able. Faults are identified and dealt with at defined instances not during produc-

tion.This then eradicates Unnecessary downtime,expensive part replacements,expensive repair bills and reduced productivity.

b. To keep machine working efficiently

machines have to be kept in good health, that is to say running efficiently and smoothly, this results in high level of productivity, hence company can meet its supply and demand, also offers reliability,reliable plant results in high profit.

c. To prevent injuries

Unmaintained machines can be dangerous to operators, as they are not in their proper functionality and their behaviour cannot be fully controlled.

1.1.2 The analysis of the different forms of maintenance is based on 4 concepts:

- The events which are at the origin of the action: reference to a timetable, relation to a type of event (self diagnosis, information from a sensor, measurement of wear, etc.), the appearance of a failure.
- The maintenance methods which will be respectively associated with them: systematic preventive maintenance, conditional preventive maintenance, corrective maintenance etc.
- The actual maintenance operations: inspection, control, troubleshooting, repair, etc.

- Related activities: improvement maintenance, renovation, reconstruction, modernization, new works, security, etc.

1.2 Types of Maintenance

a. Reactive Maintenance(RM)

Reactive Maintenance (RM) sometimes known as breakdown maintenance is a run-to-failure maintenance method. Maintenance to repair equipment only performed when equipment has broken down or been run to the point of failure. In practical a few plants or companies use a true run-to-failure maintenance management philosophy. With RM you get the maximum utilization and production output of the equipment since it's used to its limits.When a company opts for run-to-failure management, it does not spend any money on maintenance until a machine or system fails to operate. However, the cost of repairing or replacing a component would potentially be more than the production value received by running it to failure. Furthermore, as components begin to vibrate,overheat and break, additional equipment damage can occur,potentially resulting in further costly repairs. In addition, a company should maintain extensive spare inventories for all critical equipment and components to react to all possible failures. The alternative is to rely on equipment vendors that can provide immediate delivery of all required spare equipment and components.

Advantages of RM

- No initial cost: poses no initial cost as machine is only repaired after its run to failure.

- No planning involved: Technicians repair equipment when it fails. Since fails are unpredictable, no time is spent planning the repairs

Disadvantages of RM

- More expensive: unexpected failure may prove to be costly as a result of late orders for replacing parts, hence damaged reputations and impacted revenue. The unpredictable nature of RM means that labour and spare parts may not be readily available so organizations can end up paying a premium for emergency parts shipping, travel time, and after-hours support.
- Safety issues: Machines that are left to run to failure may pose as a risk to operators as their behaviour cannot be determined.
- Inefficient use of time: normally reactive maintenance catches the owners of the machine unaware and may be time consuming, as they gather information needed for reparation which may include searching for personnel with the necessary expertise and also ordering of parts which need to be replaced.
- Bad for back-log: Once things to be maintain are left to pile up this may result in maintenance backlog which ends up being hard to solve.
- Higher energy costs: Failure to maintain machinery normal results in more use of energy as most of the energy is lost due to overheating, abrasive filing of parts etc, this can be avoided by doing simple acts such as greasing machinery parts.

b. Corrective Maintenance(CM)

Corrective maintenance operations take place once the failure is identified. It is basically a troubleshooting since corrective maintenance is performed

after detection of a failure and intended to return an item to a state in which it can perform a required function.The stages of corrective maintenance are after failure, we do diagnosis, eliminate the part causing the failure then order the replacement of the part, replace the part then conduct a test of function and finally the continuation of use of the machine. The Flowchart for Corrective Maintenance is shown in figure 1.1.
Corrective maintenance is employed after an additional problem is discovered during a separate work order. For example, during one of the routine inspections, a technician realizes that there is an issue that needs corrected before other problems arise.
We have realized that corrective maintenance is performed "just in time,"hence giving a machine benefit of reduced emergency maintenance orders thereby increases employee safety, helps maintenance teams resolve problems before delays in production or service interruptions occur.
Corrective maintenance helps an organization to extend the lifespan of its equipment, reduce employee injury, and also optimize resource planning.

When to use Corrective Maintenance

- When asset failure doesn't affect safety
- When a system has redundancies that allow it to operate properly even if a part fails
- When non-critical assets can be allowed to run to failure and are inexpensive and easy to repair or replace
- When condition-based monitoring finds machine anomalies that signal potential failure
- When preventive maintenance tasks identify potential faults

Advantages of CM

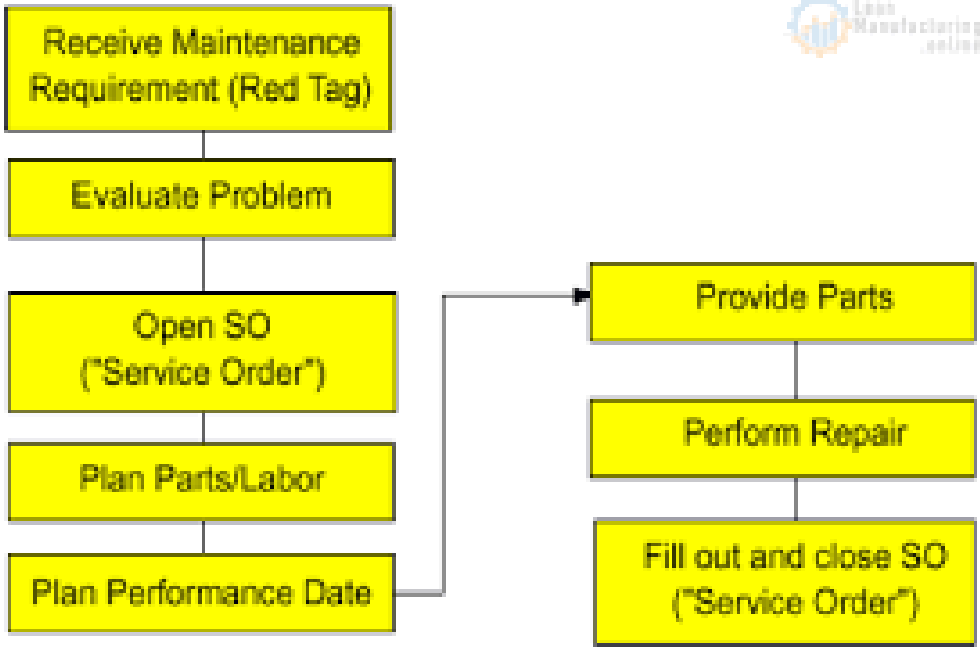


FIGURE 1.1: Corrective Maintenance Flowchart [1]

- helps maintenance teams resolve problems before delays in production or service interruptions occur
- reduced emergency maintenance orders as no piling of faults is done.
- increased employee safety as machines are repaired just in time hence always in their best performance.
- Corrective maintenance consists of very targeted action on specific components that are faulty hence requires very limited complex planning

Disadvantages

- Relying on corrective maintenance can be problematic if the equipment is not monitored after purchase, leading to more failures that are highly unpredictable and whose cause is unknown.
- Unexpected failures may be slowed down by not being able to access materials which may lead to increased periods of inactivity.

- It's a short term solution for fixing assets hence this approach does not protect or look after the equipment and therefore reduces the lifetime of the assets.

- it can be an extremely costly for the organisation and slow process to fix and may result in large periods of inactivity that may have negative effects on reputation, client satisfaction, safety and as well as on the ability to run a business efficiently and productively.

c. Predetermined Maintenance

Predetermined maintenance this not a very popular kind of maintenance which solely relies on the programs that the manufacturers provides with the machine. This type of maintenance is therefore implemented as per the information provided by the manufacturers. The maintenance department has to rely on the manufacturer's program, this creates a risk of downtime to occur which can affect productivity.
The programs are made on the basis of the knowledge of failure mechanisms derived from historical observations of the equipment.

d. Condition –based maintenance

Condition-based maintenance can be considered to be the most complicated form of maintenance to apply. This involves regular checks and a plan to avoid failures. It monitors the actual condition of the equipment and determines the necessary maintenance operations needed based on certain indicators: performance, future failures, etc. Data is gathered automatically or remotely

using a direct network connected to the equipment, this enables the maintenance team to decide on whether to control constantly or on regular intervals by comparing with the average values and performance. Maintenance is initiated when indicators show that the equipment is deteriorating and that failure probability is increasing. This type of maintenance, in the long run, helps reduce drastically the costs associated with maintenance, thereby minimizing the occurrence of serious faults and optimizing the available economic resources.Condition based maintenance Flowchart is shown in figure 1.2 Preventive Maintenance also referred to as planned maintenance,

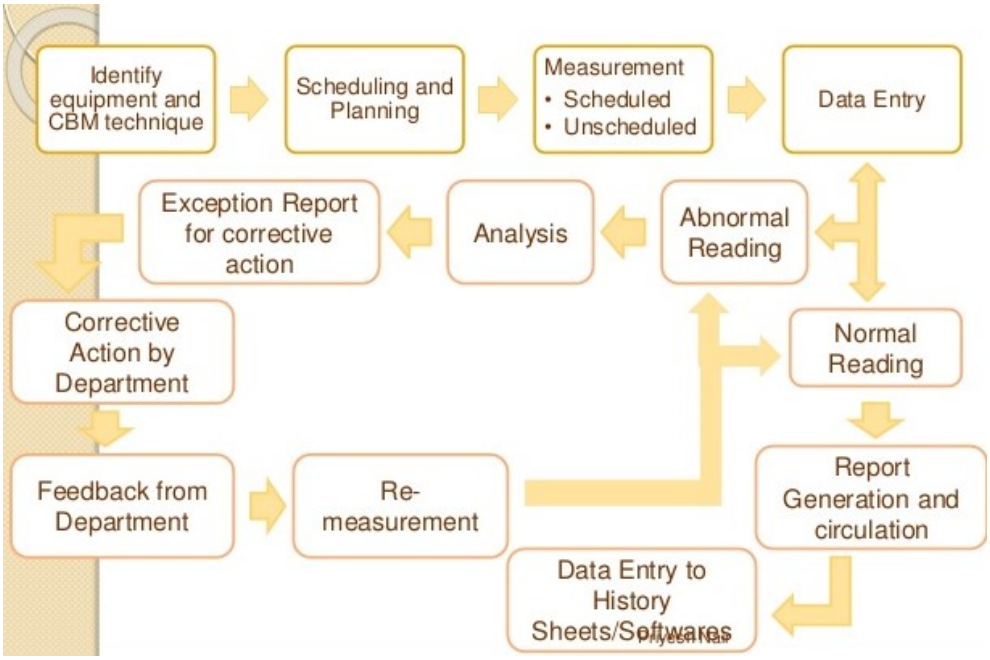


FIGURE 1.2: Condition Based Maintenance Flowchart [2]

performed at predetermined intervals or according to prescribed criteria with the intention to reduce the probability of failure or the deterioration of the operation of a good or the deterioration of a service rendered [14].The maintenance is executed even when the machine is still working and under normal operation so that the unexpected breakdowns with the associated downtime and costs would be avoided.

PM could reduce the repair costs and unplanned downtime,but might result in unnecessary repairs or catastrophic failures.Determining when a piece of equipment will enter the wear out phase is based on the theoretical rate of failure instead of actual stats on the condition of the specific equipment. This often results in costly and completely unnecessary maintenance staking place before there is an actual problem or after the potentially catastrophic damage has begun.

In a proactive approach, it takes into account several criteria to anticipate malfunctions of a good or service:regulations affecting certain materials and certain infrastructures to meet standards (in the pharmaceutical or aeronautical industry for example), user feedback and machine reports, manufacturers' recommendations, to be taken into account in particular in order to be able to apply the warranty or quality assurance in the event of a breakdown.

There are two types of preventive maintenance: Systematic preventive maintenance

It is established according to a calendar or frequency of use (every month, every 500 products manufactured, every 10,000 kilometers, etc.)

Conditional preventive maintenance

It is based on the condition, therefore the actual condition of the equipment, via continuous or regular monitoring and direct connection. The deterioration of the asset is measured by self-diagnosis, the information from a sensor, the measurement of wear, etc. It is very economical because it does not require a big management of stocks, purchases and useless interventions.

The preventive maintenance flowchart is shown in figure 1.3

Advantages of PM

- Planning is the biggest advantage of a preventive maintenance program over less complex strategies as unplanned maintenance have many

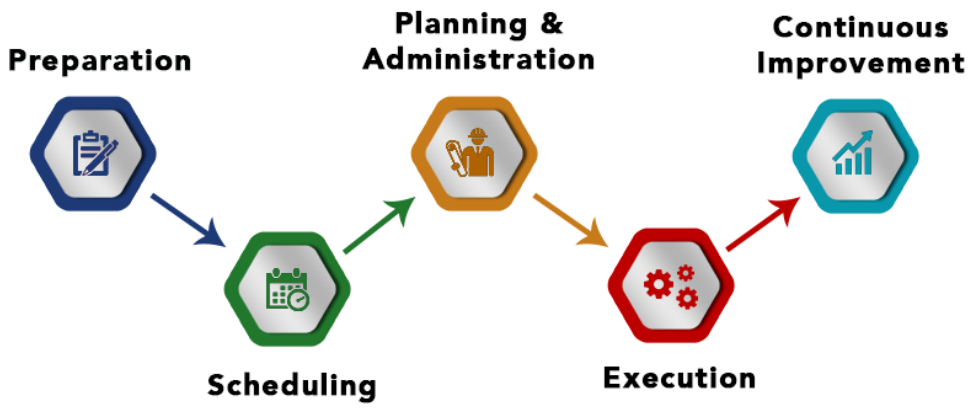


FIGURE 1.3: Preventive Maintenance Flowchart [3]

overhead costs.

- Better margins and profits due to less downtime.
- Increased safety and reduced risk of injury as machines are maintained at their best condition.

Disadvantages of PM

- Unlike reactive maintenance, preventive maintenance requires maintenance planning. This requires an investment in time and resources that is not required with less complex maintenance strategies.
- Maintenance can be done too soon hence removing a part that could still have more RUL.
- frequency of preventive maintenance is most likely to be too high

e. Prognostics/Predictive Maintenance

Prognostics is currently at the core of systems health management,the ultimate goal is to catch breakdowns before they happen by monitoring equip-

ment conditions,its major barrier is the time it takes to implement rather than the cost of the technology [15].In this type of maintenance, the condition of a system is monitored using sensor devices.IoT sensor devices have also made it possible to monitor the system remotely. Reliably estimating remaining useful life (RUL)holds the promise for considerable cost savings (for example by avoiding unscheduled maintenance and by increasing equipment usage) and operational safety improvements. RUL estimates enable decision makers to acquire information that allows them to decide on operational characteristics (such as load) which in turn may prolong the life of the component.
PdM enables organisations to reduce the frequency of unplanned RM and also helps evict incurring costs associated with PM as parts are removed too early.This concept has existed for many years, of recent emerging technologies become both seemingly capable and inexpensive enough to make PdM widely accessible [16]. PdM typically involves condition monitoring,fault diagnosis, fault prognosis, and maintenance plans [17].Technological advancements have the enhanced potential to detect, isolate, and identify faults of equipment and components. The emerging that have revolutionized PdM are the following:

- IoT : IoT makes it possible to gather huge amount of data from multiple sensors installed on machines or components [6].

- Big data techniques for data preprocessing: since a huge amount of data is collected from the machine, there is need for data preprocessing so as to turn the big machinery data into actionable information, e.g data cleaning and transforming, feature extraction and fusion, etc.

- Advanced Deep Learning (DL) methods for fault diagnosis and prognosis: continuous research and publications have resulted in better deep

learning methods that enable more accuracy of fault diagnosis and prognosis e.g RUL prediction.

Figure 1.4 outlines all the stages from start to end of a predictive maintenance workflow.

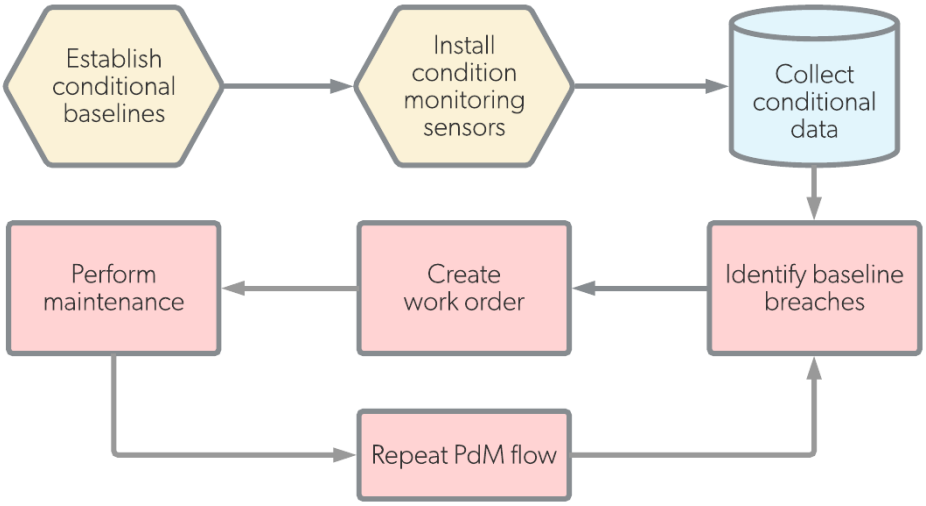


FIGURE 1.4: Predictive Maintenance Workflow [4]

Advantages of PdM

- Reduction or near elimination of unscheduled equipment.
- downtime caused by equipment or system failure.
- Increased labor utilization.
- Increased production capacity.
- Reduced maintenance costs.
- Increased equipment lifespan.

Disadvantages of PdM

- Data can be misinterpreted, leading to false maintenance requests,

- It's costly to establish a complete IoT system with sensors, transmission costs and analysis,
- Predictive analysis may not take contextual information into account, such as equipment age or weather,
- Predictive maintenance may discourage proactive physical inspection and equipment maintenance,
- Preventative maintenance activities may be triggered by timelines rather than genuine machine condition.

1.3 Machine Learning

Machine Learning is a sub-domain of artificial intelligence. It is an artificial intelligence (AI) technology that allows computers to learn without having been explicitly programmed for it. To learn and develop, however, computers need data to analyze and train on. In fact, Big Data is the essence of Machine Learning, and it's the technology that unlocks the full potential of Big Data. Machine Learning is very effective in situations where insights need to be discovered from large, diverse and changing data sets, that is, Big Data. For the analysis of such data, it proves to be much more efficient than traditional methods in terms of accuracy and speed. For example, based on information associated with a transaction such as the amount and location, and on historical and social data, Machine Learning can detect potential fraud in a millisecond. This method is therefore significantly more efficient than traditional methods for analyzing transactional data, data from social networks or Customer Relationship Management (CRM) platforms.

Machine learning remains limited, however, depending on the amount of input data that the information has. For example, to analyze a conventional size image, several thousand pixels will be sent to the machine. It will therefore be necessary to create a system for receiving and grouping information in order to be able to select those that interest the algorithm. And this is where deep learning comes in.

1.4 Deep Learning

Deep Learning is itself a subcategory of machine learning. The most common application example is visual recognition, computer vision,automatic speech

recognition,natural language processing,audio recognition and bioinformatics. For example, an algorithm will be a programmer to detect certain faces from images coming from a camera. Depending on the database obtained, he will be able to locate an individual wanted in a crowd, detect the satisfaction rate at the exit of a store by detecting smiles, etc. A set of algorithms will also be able to recognize voice, tone, the expression of questioning, an affirmation and words. To do this, Deep Learning is mainly based on the reproduction of a neural network inspired by the brain systems present in nature. The developers decide according to the desired application what type of learning they will implement. In this context, we speak of supervised learning, unsupervised learning in which the machine will feed on data not previously selected, semi-supervised, by reinforcement (linked to an observation), or by transfer in which the algorithms will apply a solution learned in a situation never seen before.
On the other hand, this technique needs a lot of data to train and obtain success rates sufficient to be used. Deep learning also requires more computing power to do its job.

Considering its wide application and potential,it is an ideal candidate to be used for system health monitoring applications. A deep learning architecture is capable to extract hierarchical representation of the data automatically and then utilised the rest of the stacked layers to learn complex features from the simpler ones.In addition,such an approach can be used to achieve an end-to-end system that can automatically learn features from its raw inputs and be able to process accordingly. In contrast to conventional machine learning, deep learning may not require extensive human interaction and knowledge for feature design. In fact, some architectures can be used to learn a model of the input distribution from which one can generate samples. They can also be seen as unsupervised feature learning algorithms and

hence be used to pre-train features from labelled or unlabelled data. Such features can then be used as initialisation for supervised networks.

1.4.1 Types of Machine Learning Algorithms

a. Supervised Machine Learning Algorithms

Supervised machine learning algorithms are the most commonly used. With this model, a data scientist serves as a guide and teaches the algorithm the conclusions it must draw. Examples of supervised machine learning include algorithms such as classification , linear and logistic regression, and machines with support vectors.
In this mode of machine learning, there is no question of relying on predefined elements, and the task is for the machine to proceed by itself to categorize the data. To do this, the system will cross the information submitted to it, so as to be able to gather in the same class the elements having certain similarities. Thus, depending on the goal sought, it will be up to the operator or the researcher to analyze them in order to deduce the different hypotheses.
Just as a child learns to identify different types of fruits by memorizing them from a picture book, in supervised learning, the algorithm learns thanks to a data set already labeled and whose result is predefined [18].

In figure 1.5 labeled data is fed into a machine learning algorithm producing a model, with the training data the machine adjusts itself by changing parameters thereby constructing a logical model.In figure 1.6 this built model is then fed with a new set of data to predict outcome.

b. Unsupervised Machine Learning Algorithms

Unsupervised machine learning uses a different approach, input is data without labels or specific defined results, a computer learns to identify on

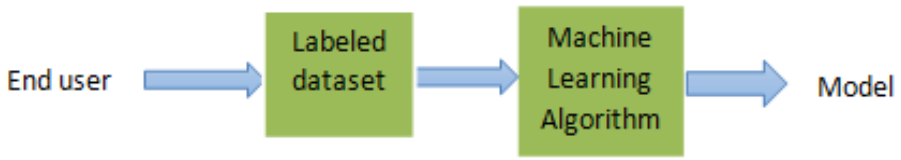


FIGURE 1.5: Supervised Machine Learning Step1

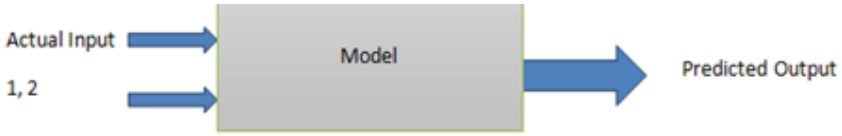


FIGURE 1.6: Supervised Machine Learning Step2

it's own complex the similarities and distinctions within these data, and to group together those which share common characteristics without any constant and rigorous human guidance. Examples of unsupervised machine learning are principal component , association rule and cluster analysis. Unsupervised machine learning can be compared to a child who gets to learn to identify the type of fruit by observing pattern and color, instead of memorizing names with the help of one another person. He/She looks for similarities between the images hence separates them into groups, thereby assigning each group its own label [18]

.

In figure 1.7 data is fed to the machine learning algorithm which then predicts the outcome as a class label in which the input belongs,clusters are formed by finding similarities among the inputs.

c. Reinforcement Machine Learning Algorithms

Reinforcement Machine Learning works using feedback, as shown in fugure 1.8, system learns by trial and error using information from its previous

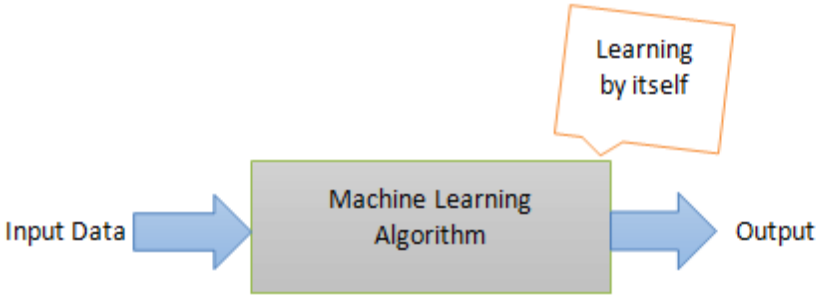


FIGURE 1.7: Unsupervised Machine Learning

actions and experiences. Reinforcement learning is an iterative process. The accuracy is dependent on the number of feedback.

A good example of reinforcement learning is when a child is playing video games where the players complete certain levels of a game and earn reward points. The game provides feedback to this child through bonus moves to improve his/her performance.Normally used to train self driving cars,robots etc.

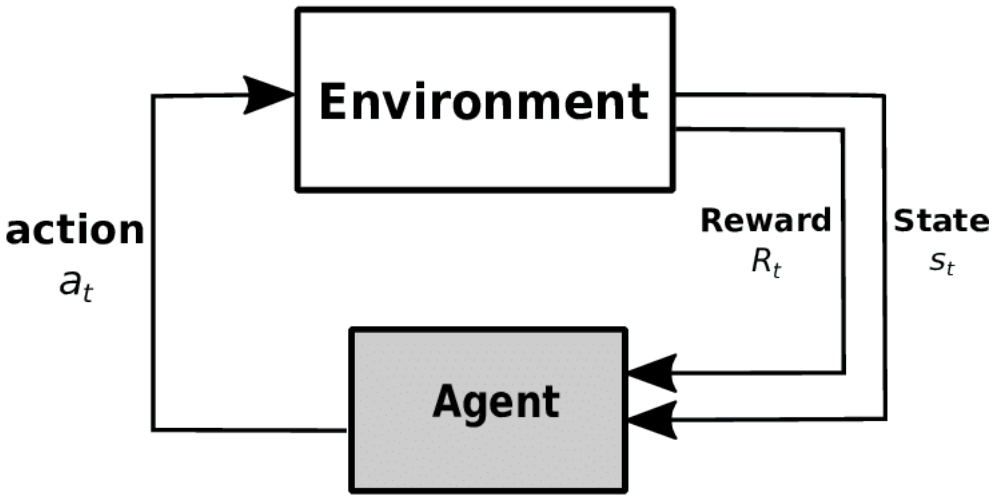


FIGURE 1.8: Reinforcement Learning, Agent and Environment [5]

1.4.2 Recurrent Neural Networks(RNN)

The RNN is a framework used to deal with sequential data, hence its capabilities for health management systems due to their time series nature. RNN overcome limitations of simple neural nets by using information from the past network results to produce the output as illustrated in 1.9. There exists many RNN variations; the two more popular ones are networks that incorporate Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU). These variations were introduced to address the vanishing or exploding gradient problem in RNNs. Some authors advocate that these variations are better than traditional RNNs due to their ability to store long-term dependencies and non linear dynamics in time series data; making them ideal for time series sensor, data processing and health monitoring.

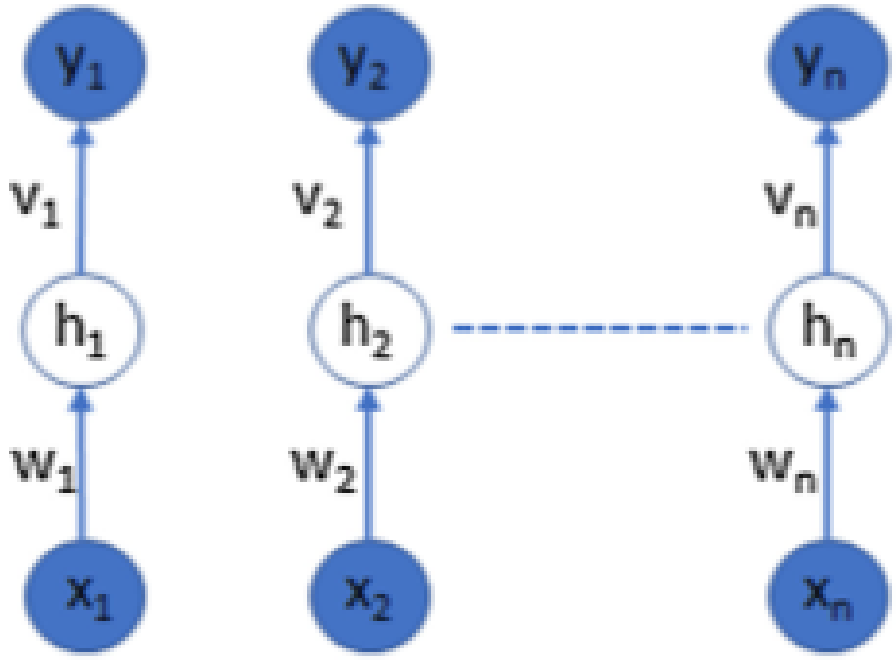


FIGURE 1.9: Sequential structure independence in a simple neural network [6]

- x_i : input state.
- y_i : output state.
- h_i : hidden state.
- v_i and w_i : weights of network.

1.4.3 Application of Deep learning

Deep Learning is used in many fields:

- image recognition,
- automatic translation,
- autonomous car,
- medical diagnosis,

- personalized recommendations,
- automatic moderation of social networks,
- financial prediction and automated trading,
- identification of defective parts,
- detection of malware or fraud,
- chatbots (conversational agents),
- space exploration,
- intelligent robots.

1.5 Previous Papers review

Rui Zhao et Al in 2016 [19] used LR, SVR, MLP, RNN, Basic LSTMs, Deep LSTMs.A comparison is conducted of LSTMs with several benchmark methods, MAE and RMSE of all methods are noted. They deduced that among regression models including LR, SVR and MLP based on expert features, LR had the worst performance, the reason being the limitation of linear models. SVR with RBF kernel and MLP models, these models based on expert features all under-perform compared to LSTMs models. LSTMs models all work on raw signals instead of expert features. Especially, deep LSTMs achieve a significant performance gain compared to these models. Basic LSTMs perform slightly better than RNN. The reasons may be the fact that gate functions employed in LSTMs can enable it to capture long-term dependency better than RNN. Among all these models, deep LSTMs were proved to achieve the best and robust performance. Compared to basic LSTMs, deep LSTMs stacked three LSTM layers and is more capable to learn robust and abstract representations from raw data .

In 2017 Rui Zhao et Al [20] repeated the same research only that this time, they proposed CBLSTM model in addition to the models used in 2016. It is shown that the proposed CBLSTM model achieves the best performance among all compared methods. Compared to the most competitive model, deep BLSTMs, CBLSTM adopts convolutional neural network to address the raw signal and then builds recurrent modules on top of CNN. The CNN is adopted to extract local features, which can filter the noise in the raw signal effectively. In addition, CNN can also reduce the length of sequential data .

Shuai Zheng et Al 2017 [21] proposed a deep learning approach for Remaining Useful Life (RUL) estimation which they showed its benefits by taking sequence information when estimating RUL. The approach involved sequence of layers of LSTM followed by feed forward networks. They did experiments on 3 widely used data sets, C-MAPSS Data Set, PHM08 Challenge Data Set and Milling Data Set. The experiments showed that the LSTM model outperforms other approaches and gives the best performance in RUL estimation.

Long Wen et Al 2017 [22] used a new CNN based on LeNet-5 for fault diagnosis using 3 different data-sets the motor bearing data set,self centrifugal pump data set and the Axial piston hydraulic pump data set,and achieved prediction accuracy of 99.79%, 99.481%, and 100%, respectively.This method is compared with other deep learning methods : Support Vector Machines (SVM), multi-objective deep belief networks ensemble (MODBNE), Deep ANN (DNN), LSTM, Deep CNN (DCNN) , Deep LSTM and Stacking Ensemble. The results showed that the proposed method showed significant improvement

Xiang Li et Al 2018 [23]used DCNN method to determine the RUL using the same Turbofan NASA Data set. The method was then compared with the following methods for estimation of RUL, basic neural network (NN), deep neural network (DNN), recurrent neural network (RNN) and long short-term memory (LSTM).The proposed deep CNN was seen to be more suit for RUL prediction as it gave better accuracy, with RNN being the second best and the rest did not give satisfactory results.

Yanting Zhou et Al 2019 [24], this paper studies the influence of temperature and voltage on the aging trend of supercapacitors combined with the degradation mechanism model of supercapacitors so as to evaluate the reliability of supercapacitor system and determine possible failure times in advance.The methods applied are the LSTM RNN,GRU and the SIM RNN. It's proved that LSTM RNN performs high prediction accuracy and good robustness. Moreover, the proposed method is compared with GRU and SIM RNN for the prediction of offline data, which verifies the validity and applicability of the proposed method.

Conclusion

In this subsection, we summarize the types of maintenance strategies in benefits, challenges, suitable applications. It can be found that RM has the lowest prevention cost due to using run-to-failure management, PM has the lowest repair cost due to well scheduled downtime while PdM can achieve the best trade-off between repair cost and prevention cost. Ideally, PdM allows the maintenance frequency to be as low as possible to prevent unplanned

RM, without incurring costs associated with doing too much PM. Note that prevention cost mainly contains inspection cost, preventive replacement cost, etc., while the repair cost denotes the corrective replacement cost after failure occurred. After analysis we realize that predictive maintenance has more advantages if it is adopted as compared to all other forms of maintenance. Finally I looked at papers published in the past 5 years, I took note of the methods used for RUL predictions.From all this we see a very significant improvement as years go by in the methods used for RUL estimation propagated by continous scientist research.All the papers reviewed either use Convolutional Neural Network (CNN), a Deep Belief Network (DBN) or Long-Short Term Memory (LSTM) in the proposed deep architecture.The method that gave the best results is the LSTM.

Chapter 2

Case Study

Introduction

In this chapter we are going to look at our data set, how it was acquired, the different elements that form our data-set. So we are going to summarize the different modeling methods used to acquire the data.Furthermore we will look at the different model architectures applied.

2.1 Software and Hardware Environment

- Experiments done on a personal computer with Intel core i7-4500U(2.64GHz) CPU, 16GB RAM, and Windows operating system.
- All codes written on python 3.7 with "tensorflow" library and deep learning library "Keras"

2.2 Modeling Of Data-set

2.2.1 System Modeling

A simulation of the turbofan engine was done using the C-MAPSS (Commercial Modular Aero Propulsion System Simulation), coding on this software is coded in the MATLAB, in collaboration with the Simulink library. The software possesses a number of editable input parameters, which enable

user to change operational conditions. The engine diagram in figure 2.1 shows the main elements of the engine model and the flow chart in figure 2.2 shows how various subroutines are assembled in the simulation.

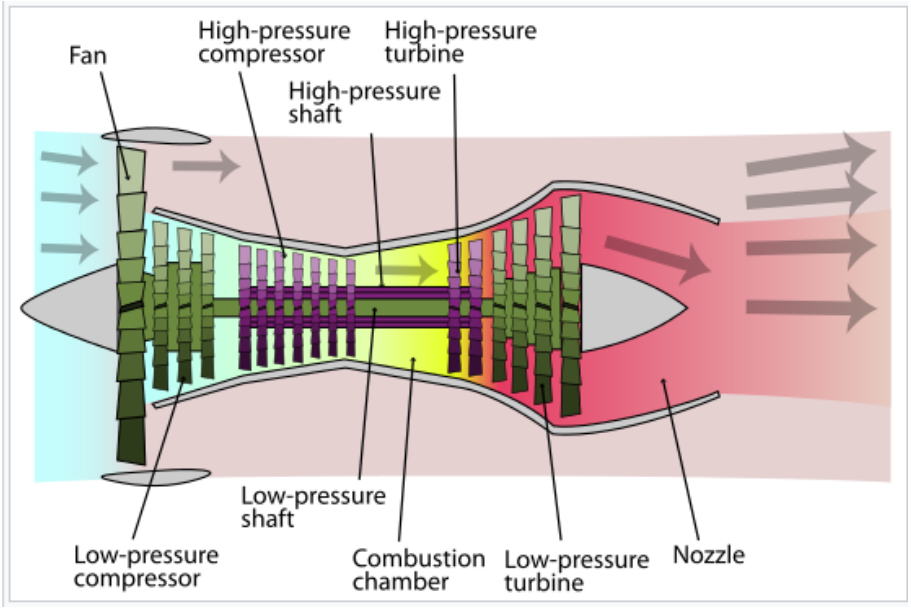


FIGURE 2.1: Simplified diagram of engine simulated in C-MAPSS [7]

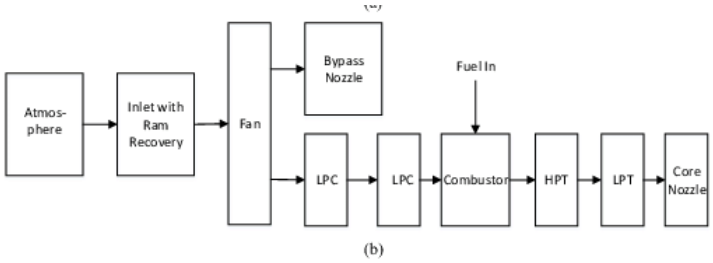


FIGURE 2.2: A layout showing various modules and their connections as modeled in the simulation [8]

- LPC : Low pressure compressor
- HPT : High pressure turbine
- LPT : Low pressure turbine

CMAPSS can be operated either in open-loop (without any controller) or in closed loop (with the engine and its control system) configurations. For this research a closed loop configuration was used [25].

2.2.2 Damage Propagation Modeling

After getting the system model, we now have to model the propagation of damage. Common models used across different application domains include the Arrhenius model, the Coffin-Manson mechanical crack growth model, and the Eyring model. The details of these models will not be discussed here.

a. Specifications of Data Set

The data sets consist of several multivariate time series that contain subsets of the training, test, and ground truth data sets. Each time series comes from a different engine, that is, the data can be considered as coming from a fleet of engines of the same type. Each engine starts with different degrees of initial wear and manufacturing variation unknown to the user. This wear and variation is considered normal; it is not considered a fault condition. Three operational parameters have a substantial effect on engine performance. These parameters are also included in the data. Data is contaminated with sensor noise.

The engine operates normally at the start of each time series and develops a fault at some point during the series. In the training set, the fault increases in magnitude until the system fails. In the test set, the time series ends some time before the system fails.

unit-time	cycle	setting1	setting2	setting3	s1	s2	s3	....	s21

TABLEAU 2.1: Colonnes of our dataset

The training data consists of several multivariate time series with "cycle" as the unit of time, as well as 21 sensor readings for each cycle. Each time series can be assumed to be generated from a different engine of the same type. Each row is a snapshot of the data taken during a single operational cycle, each column is a different variable as shown in table 2.1. The test data has the same data schema as the training data. The only difference is that the data does not indicate when the failure occurs. Finally, the ground truth data provides the number of remaining working cycles for the engines in the test data.

b. Data Preparation

Feature scaling is an important preprocessing step in machine learning. Many machine learning algorithms use some sort of an optimization algorithmin order to adjust the feature weights. Such optimization algorithms might be gradient descent or the Levenberg-Marquardt algorithm [26]. Originally, the different features in the data-set often range between different values and have different units. For instance, there could be a data-set with a feature measuring temperature, typically between -30 to 40 degrees, and then have another feature measuring pressure within the range of 500 to 2000 Pa. The difference in the range of values of these features is large, and without any form of scaling, it might result in certain feature weights updating faster than others. This will then lead to a lower prediction accuracy [27]. There is mainly two ways to scale the data:

- Standardization, is when the features are scaled in a way that they

have the properties of a standard normal distribution with mean value = 0 and standard deviation = 1. When performing this standardization, the creator is assuming that the data-set fits the Gaussian distribution with a well behaved mean and standard deviation. It is possible to standardize a data-set that doesn't have this assumption, but the result might not be reliable.

- Normalization, is when the features are scaled to a fixed range, often [0,1] or [-1,1].

Due to the heterogeneous data, normalization has been performed on the data-set, ranging it's values from 0 to 1. Many machine learning approaches, and especially neural networks are sensitive to the input data range, making the scaling step a necessity [28]. In real data applications, enormous raw sensor data, operating parameters, loads, and failure times will be given. Due to the fact that the Variables that are measured at different scales do not contribute equally to the model fitting and model learned function, this might end up creating a bias, sensor data will need to undergo normalization with respect to each sensor before training and testing. The MinMAx normalization is performed to our data-set. Normalization is used to scale the data between 0 and 1, meaning that the minimum and maximum value of a feature/variable is going to be 0 and 1, respectively. In this technique of data normalization, linear transformation is performed on the original data. Minimum and maximum value from data is fetched and each value is replaced according to the following formula.

$$Y_i = \frac{X_i - min(X)}{max(X) - min(X)}$$

Where X_i is the iþdata point and min represents the minimum and Maximum represents maximum. So Xi converts to Y_i

2.3 Models used

2.3.1 Model Configuration

- Error metric: Mean-Squared Error
- Number of epochs: The number of epochs refers to how many times the model has trained on the training set. 13 epochs were used in this research.
- Batch size: The entire dataset is not computed in the network at the same time. It is partitioned into batches of 200 samples. This means that after every 200th sample a back-propagation is performed and feature weights are updated.
- Optimizer: The ADAgrad optimizer is used for the regression problem, while the ADAM optimizer is used for the classification problem.
- Activation function: The ReLU activation function is used for the regression problem and the sigmoid activation function is used for the classification problem.
- Dropout: A technique that randomly chooses and deactivates a given percentage of the nodes in the hidden layers. It is commonly used to reduce model overfitting, and as described in chapter 4.3.3, it can be used to implement Monte Carlo dropout.
- Validation set: 20% of the training set is split into a validation set.

In this research different forms of recurrent neural cells have been used, which include the basic Recurrent Neural Network (RNN) cells,Gated Recurrent Units (GRU) cells and Long Short Term Memory (LSTM) cells.To prevent models from over-fitting, we take droupout and training process

early stopping mechanism.Probability of dropping out neurons at output are set at 0.5. The Adam optimizer is used to carry out experiments.

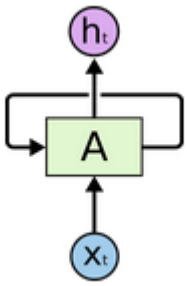


FIGURE 2.3: Recurrent Neural Network [9]

- x_t : input
- h_t : output
- A : Adam optimizer

2.3.2 Gated Recurrent Units (GRU)

GRU is a variant RNN architecture, that uses gating mechanisms to control and manage the flow of information between cells in the neural network [18],developed in 2014 by Yoshua Bengio et al are a variation of RNN cells that are easier to train while avoiding the vanishing gradients problem [29, 30],were simply created to solve the short term memory problem [31].The special attribute of a GRU is that it can be trained to keep old information, without degrading it through time or removing relevant information necessary for prediction [31]. The vanishing gradient problem arise when the gradient becomes too small, which in-turn prevents the weight from changing its value. This means that it becomes harder to the model to learn long-term dependencies in the input time-series [32, 33].This problem can solved by using either LSTM or GRU in place of the basic RNN cell which deliver promising results in many sequence learning tasks through sophisticated network designs [30].
GRUs possess one hidden state which is transferred between time steps. The hidden state has the ability to hold onto both the long-term and short-term dependencies simultaneously due to it's gating mechanisms which are trained to selectively filter out any irrelevant information,only guarding useful one. These gates are important vectors and contain values between 0 to 1 which will be multiplied with the input data and/or hidden state, if the corresponding data is important a one value in the gate vector means that the corresponding data is important and will be used.

GRU cells are used to build this model with number of epochs and the number of layers being varied, so as to carefully note the influence of the changes to the accuracy and precision of the model.

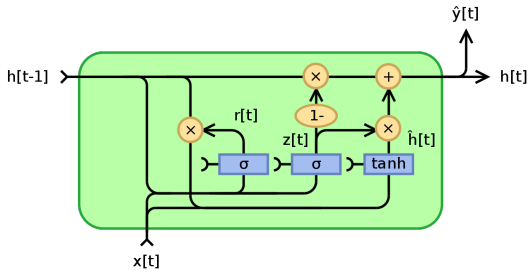


FIGURE 2.4: Gated Recurrent Network(GRU) [10]

Variables :

- x_t: input vector
- $\hat{h}_t$: candidate activation vector
- z_t: update gate vector
- r_t: reset gate vector

Activation functions:

- σ_g: The original is a sigmoid function.
- ϕ_h: The original is a hyperbolic tangent.

2.3.3 Long Short Term Memory(LSTM)

Learning to store information over long extended intervals via recurrent back propagation takes a very long time mostly due to insufficient,decaying error back-flow,the problem is addressed by using a gradient based method which is called LSTM [34].
LSTM networks are a form of recurrent neural network which is capable of learning order dependence in sequence prediction problems and this is very essential in complex problem domains like machine translation, speech recognition, and many others [35].

LSTM advantages include the ability to bridge very long time lags as a result of the constant error back propagation within memory cells,as well as not needing parameter fine tuning hence work over broader range of parameters such learning rate,input gate bias,output gate bias [34].
An LSTM layer comprises of a set of recurrently connected blocks called memory blocks. Each block contains one or more recurrently connected memory cells, as well as three multiplicative units – the input, output and forget gates, which are responsible for providing continuous analogues of write, read and reset operations for the cells. The net is designed to only interact with the cells via the gates [35].

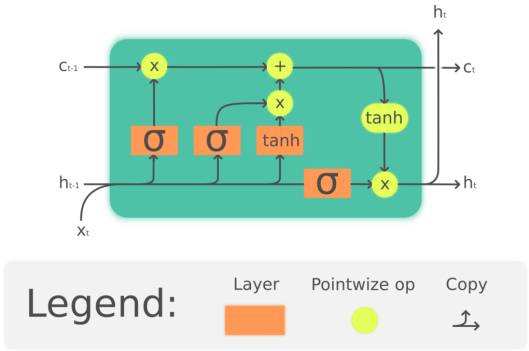


FIGURE 2.5: Long Short Term Memory (LSTM) [11]

Variables:

- x_t: input vector to the LSTM unit
- $f_t \in \mathbb{R}^h$: forget gate's activation vector
- $h_t \in \mathbb{R}^h$: hidden state vector also known as output vector of the LSTM unit
- $c_t \in \mathbb{R}^h$: cell input activation vector
- $c_t \in \mathbb{R}^h$: cell state vector

  Activation Function

- σ: sigmoid function.

2.3.4 Simple Recurrent Network(SRN)

The Simple Recurrent Network (SRN) is believed to have been first used by Jeff Elman. The SRN is a specific type of back-propagation network with feed-forward architecture, contains units in input, hidden, and output pools,as well as special type of hidden layer called a “context” layer. Here there is simple multiplication of Input and Previous Output. Passed through Tanh activation function. No Gates present.

> "The beauty of the SRN is its simplicity. In fact, it is really just a three-layer, feed-forward back propagation network. The only proviso is that one of the two parts of the input to the network is the pattern of activation over the network’s own hidden units at the previous time step." [36]

‘

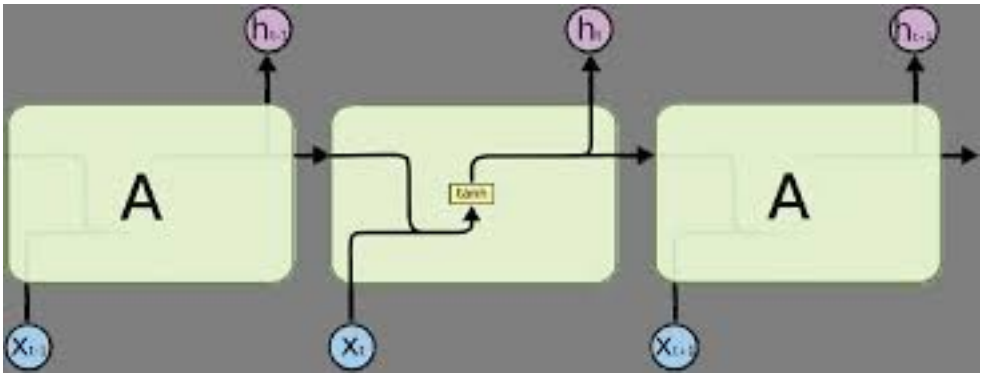


FIGURE 2.6: Simple Recurrent Neuron(SRN) [9]

- x_t: input
- h_{t-1}: previous Output
- h_t: ouput

2.3.5 The Use of CuDNN

The introduction of Powerful hardwares for complex computing as a result of rapid development of semiconductor technology, the powerful hardwares, such as graphics processing unit (GPU) and tensor processing units (TPU), has greatly influenced the significantly reduction of time of execution of Deep Learning(DL) algorithms. For example, Sunet al [37] achieved a 95-epoch train. After training using the normal LSTM and GRU cells, then changed and used CuDNN versions which are the CuDNNLSTM and CuDNNGRU known for being quick at processing large data"`...the area of Deep Neural Networks (DNNs). The success of DNNs has been greatly accelerated by using GPUs, which have become the platform of choice for training large, complex DNN-based ML systems`"[38].

> "... DNNs and CNNs require large amounts of computation, especially during the training phase.... DNNs require a large amount of training data to achieve high accuracy, meaning hundreds of thousands to millions of input samples will have to be run through both a forward and backward pass. Because neural networks are created from large numbers of identical neurons they are highly parallel by nature. This parallelism maps naturally to GPUs, which provide a significant speed-up over CPU-only training..." [38]

The speed of execution was noted and tabulated in table 3.6.

Conclusion

A case study was completed in order to test and evaluate different techniques reviewed in the literature. Data was required, and the PHM challenge looked like a suitable approach. We would have wanted to test the same methods on real data, but we faced challenges acquiring it. As mentioned in chapter **??**, the task was to calculate the remaining life of aircraft engines. The resulting case study implemented multiple deep learning methods, evaluated their performances and to some extent described the confidence level of the predictions. Although, the collected results weren't particularly impressive,which was as a result of our computational limitation, valuable experience with the implementation of data-driven prognostic models was obtained.

Chapter 3

Results and Analysis

Introduction

This chapter focuses on the experimental results obtained, will further compare between results of different models.The results were obtained after application of the LSTM,GRU and SRN models to the Nasa Turbofan data-set. Furthermore the CuDNN (CuDNNLSTM and CuDNNGRU) versions are applied and finally stacking of the GRU and LSTM cells is performed. Some of the results were due to variation in the number of epochs and layers. The activation function used: sigmoid,the optimizer: Adam and the loss function: binary cross entropy.

3.1 Long Short Term Memory(LSTM) Model

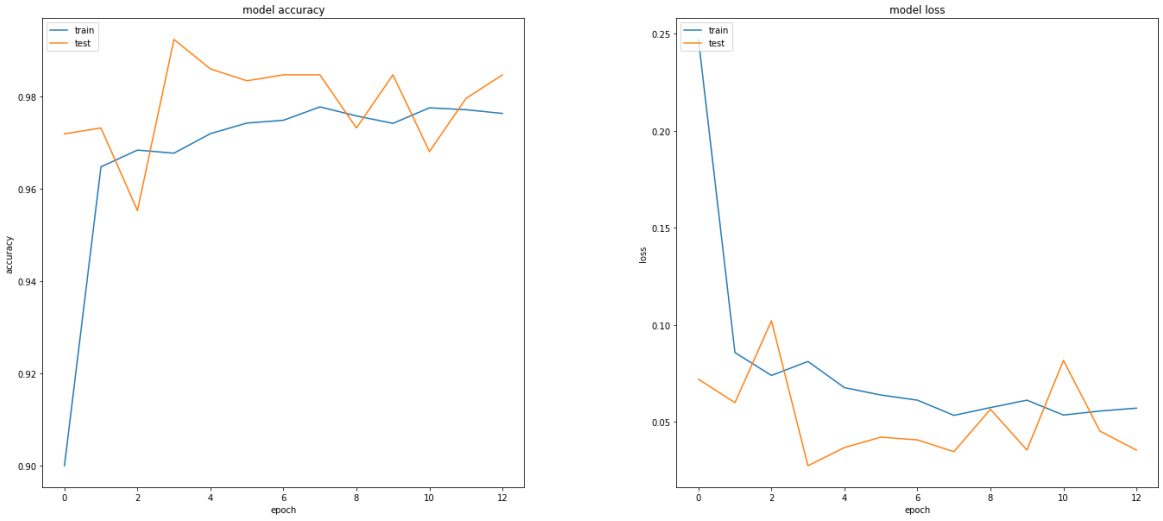
The results for LSTM model with 2 layers and 13 epochs. Figure 3.1a shows the accuracy of the model, 2 curves are used to show this accuracy, the first being accuracy of training (blue) and the second one is the test (orange). We remark that for training, the model starts directly at 90% and rises to approximately 98% ,its average is approximately 93%. For test, we realize that the curve is a bit unstable starting at approximately 97% with maximum reaching almost 99.5%, the average still maintains at approximately 93%. Training of this method is good enough.
Figure 3.1b shows the model loss, for training (blue curve), the loss starts at approximately 25% and decreases up to almost 5%, on average it is between

5% and 10%. In test (orange curve), the loss starts at around 7% decreases to approximately 4%, with maximum approximately 10% , minimum of about 2% and having an average of around 5%, it is highly unstable though it decreases better than the one for training.

Figure 3.2 is a comparison of the predicted outcome (blue) and the actual outcome (red), we can observe that in most parts of the given range, the curve in blue is totally superposed with the curve in red, hence excellent prediction which confirms the accuracy (0.9802) and precision (0.9583) shown in table 3.1, with the exception of just a small portion where the predicted value deviates a bit from the actual value.

Number of Epochs	Number of layers	Accuracy	Precision	Time of Execution(s)
13	2	0.9802	0.9583	162.7

TABLEAU 3.1: LSTM model results with 13 epochs and 2 layers



(A) Model Accuracy (B) Model Loss

FIGURE 3.1: Figures for the LSTM Model with 13 epochs and 2 layers.

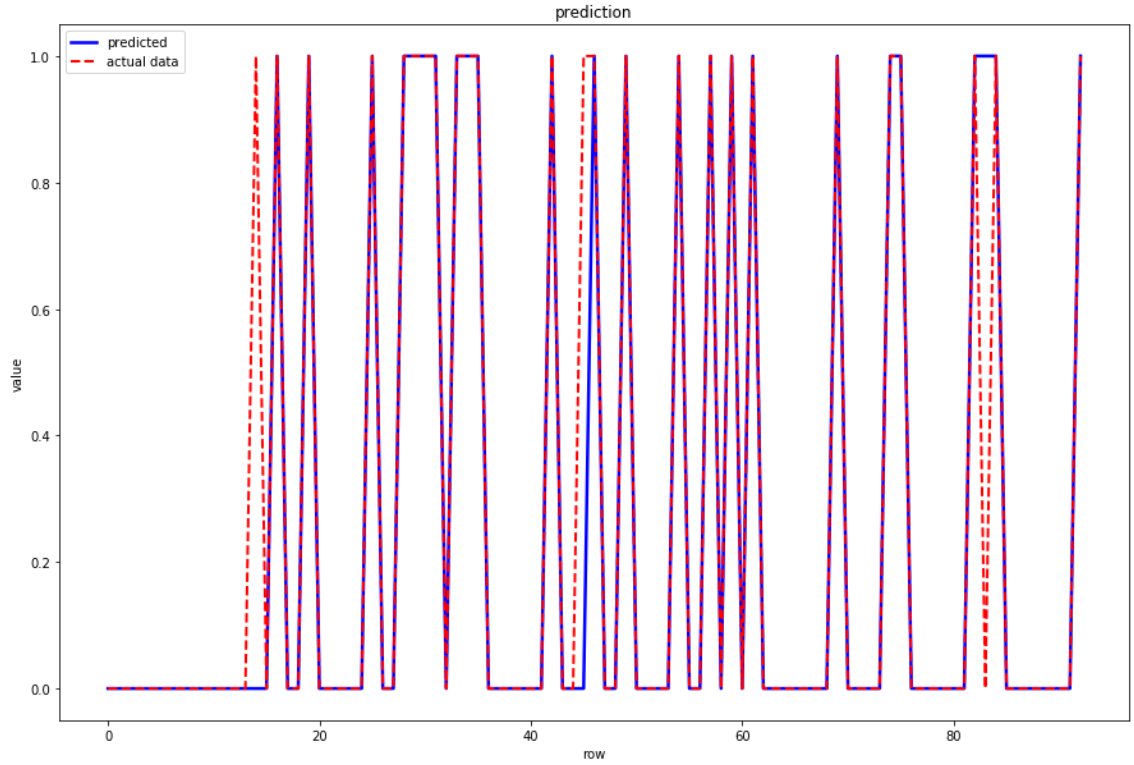


FIGURE 3.2: Predicted(_) and Actual data(_)

3.1.1 CuDNNLSTM

The results for CuDNNLSTM model with 2 layers and 13 epochs. Figure 3.3a shows the accuracy of the model, 2 curves are used to show this accuracy, the first being accuracy of training (blue) and the second one is the test (orange). We remark that for training, the model starts at around 89% and rises to approximately 98% ,its average is approximately 92%. For test, we realize that the curve is a bit unstable (more unstable than the LSTM Model above) starting at approximately 94% with maximum reaching almost 99%, the average still maintains at approximately 92%. Training of this method is good enough.

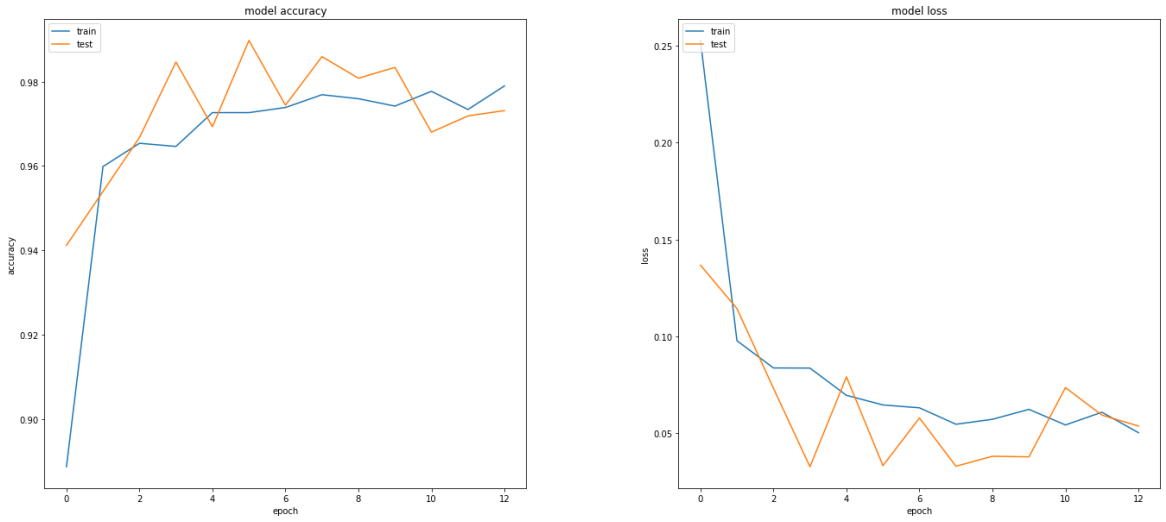
Figure 3.3b shows the model loss, for training (blue curve), the loss starts at

approximately 25% and decreases up to almost 5%, on average it is between 5% and 10%. In test (orange curve), the loss starts at around 14% decreases to approximately 5%, with minimum of about 2% and having an average of around 5%, it is highly unstable though it decreases better than the one for training.

Figure 3.4 is a comparison of the predicted outcome (blue) and the actual outcome (red), we can observe that in most parts of the given range, the curve in blue is totally superposed with the curve in red, hence excellent prediction which confirms the accuracy (0.9717) and precision (0.8799) shown in table 3.2, with the exception of just a small portion where the predicted value deviates a bit from the actual value.The accuracy for the cuDNNLSTM is almost the same as that of LSTM, though it's precision is very low compared to the one of LSTM, nevertheless cuDNNLSTM is preferable because of its less time of execution (33.02s) as compared to the 142.8s of LSTM.

Number of Epochs	Number of layers	Accuracy	Precision	Time of Execution(s)
13	2	0.9717	0.8799	33.02

Tableau 3.2: CuDNNLSTM model results with 13 epochs and 2 layers



(A) Model Accuracy (B) Model Loss

FIGURE 3.3: Figures for the CuDNNLSTM Model with 13 epochs and 2 layers.

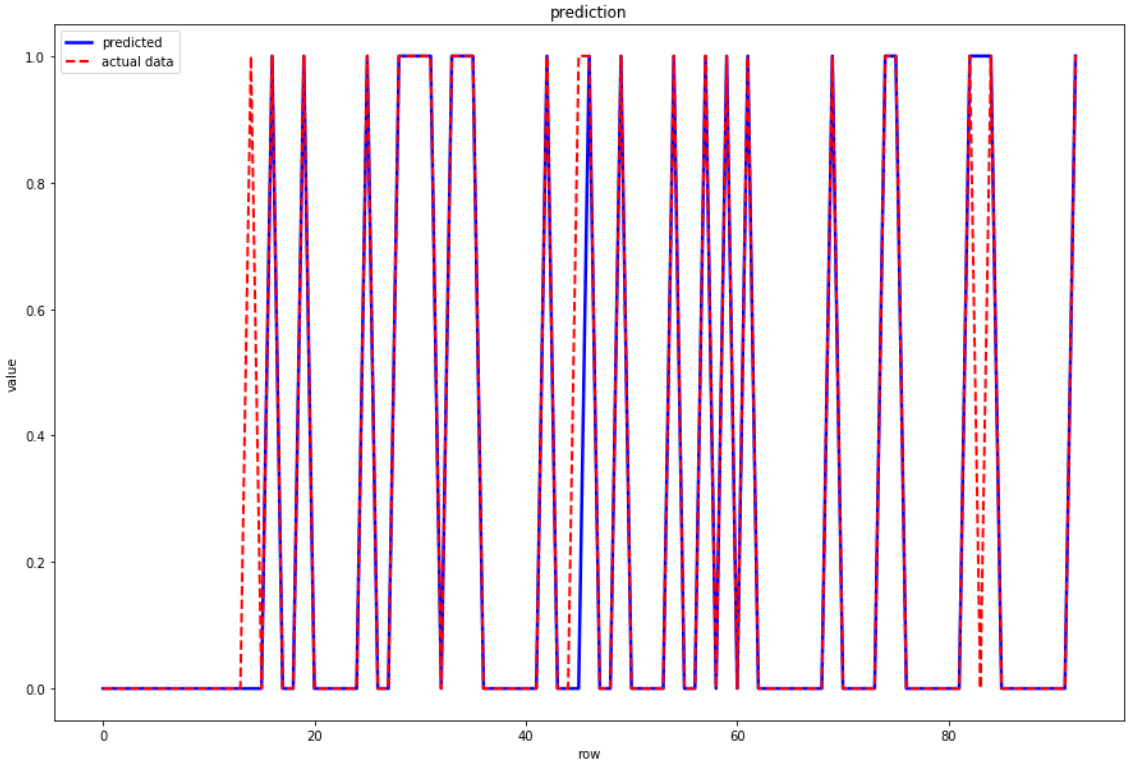


FIGURE 3.4: Predicted(_) and Actual data(_)

3.2 Gated Recurrent Unit(GRU) Model

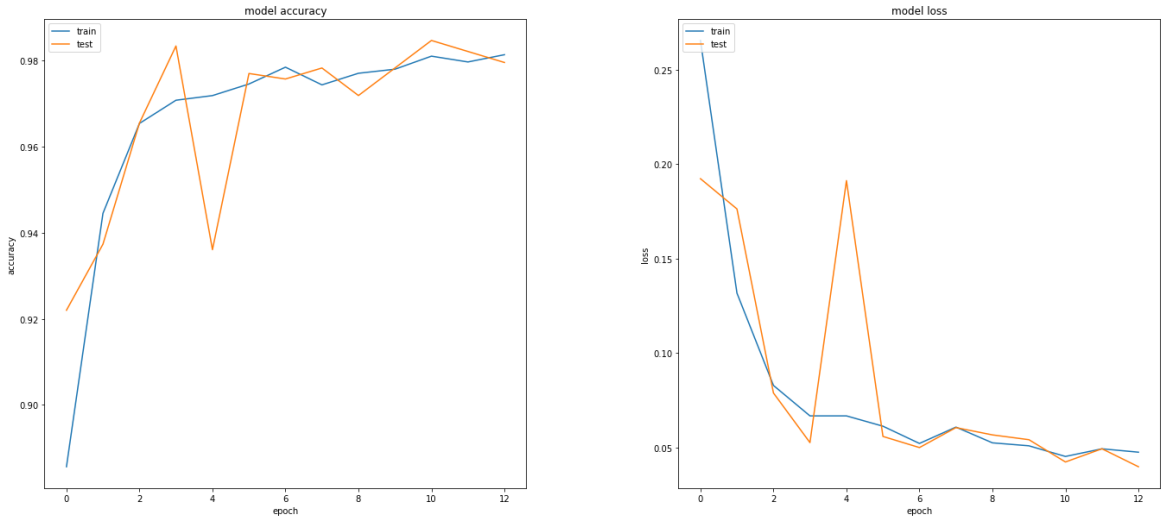
The results for GRU model with 2 layers and 13 epochs. Figure 3.5a shows the accuracy of the model, 2 curves are used to show this accuracy, the first being accuracy of training (blue) and the second one is the test (orange). We remark that for training, the model starts at 88.5% and rises to approximately 98% ,its average is approximately 93%. For test, we realize that the curve is a bit unstable starting at approximately 92% increases to approximately 98% with maximum reaching almost 98.5%, the minimum 93.5% and the average approximately 96%. Training of this method is also good enough.

Figure 3.5b shows the model loss, for training (blue curve), the loss starts at approximately 25% and decreases up to almost 5%, on average it is between 5% and 10%. In test (orange curve), the loss starts at around 20% decreases to approximately 4% (min) and having an average of around 10%, it is moderately unstable.

Figure 3.6 is a comparison of the predicted outcome (blue) and the actual outcome (red), we can observe that in most parts of the given range, the curve in blue is totally superposed with the curve in red, hence excellent prediction which confirms the accuracy (0.9829) and precision (0.9433) shown in table 3.3, with the exception of just a small portion where the predicted value deviates a bit from the actual value.

Number of Epochs	Number of layers	Accuracy	Precision	Time of Execution(s)
13	2	0.9829	0.9433	149.58

Tableau 3.3: GRU model results with 13 epochs and 2 layers



(A) Model Accuracy (B) Model Loss

FIGURE 3.5: Figures for the GRU Model with 13 epochs and 2 layers.

3.2.1 CuDNNGRU

The results for cuDNNGRU model with 2 layers and 13 epochs. Figure 3.7a shows the accuracy of the model, 2 curves are used to show this accuracy, the first being accuracy of training (blue) and the second one is the test (orange). We remark that for training, the model starts at 90% and rises to approximately 98% ,its average is approximately 96%. For test, we realize that the curve is a bit unstable starting at approximately 98% decreases to 97% with maximum reaching almost 99.5%, the average approximately 98%. Training of this method is also good enough.

Figure 3.7b shows the model loss, for training (blue curve), the loss starts at approximately 25% and decreases up to almost 5%, on average it is between 5% and 10%. In test (orange curve), the loss starts at around 5% increases to approximately 7% (max) and having an average of around 6%, it is unstable.

Figure 3.8 is a comparison of the predicted outcome (blue) and the ac-

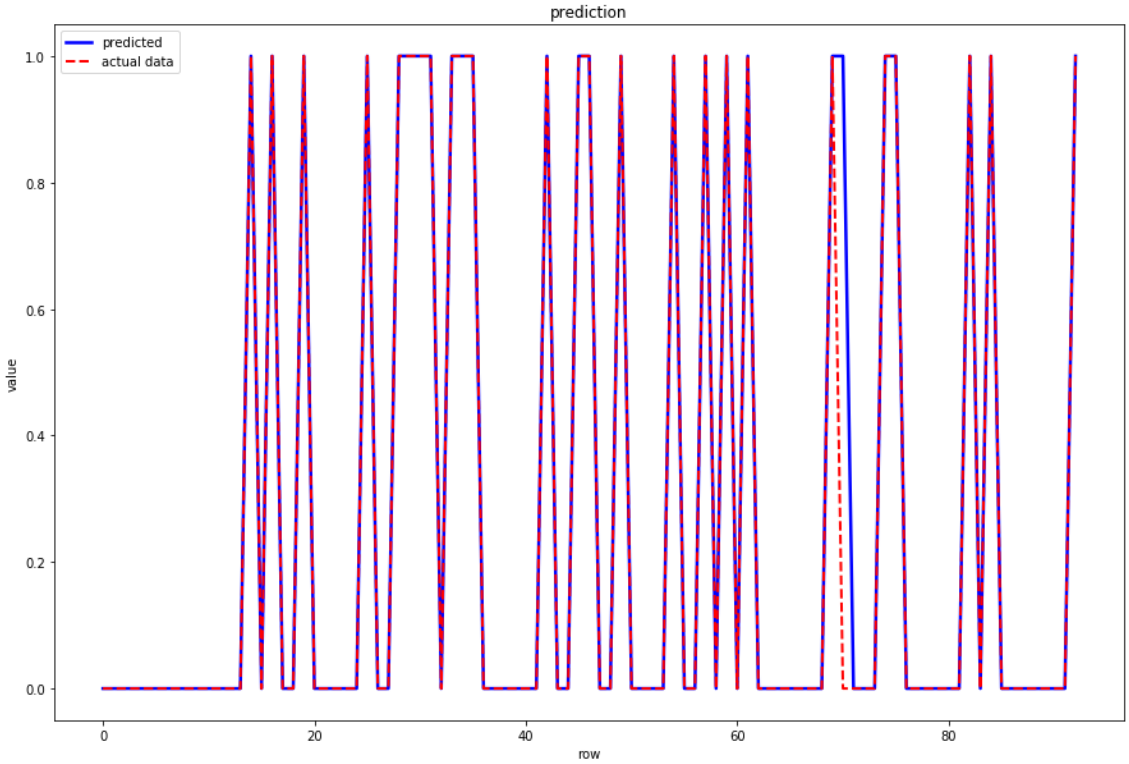
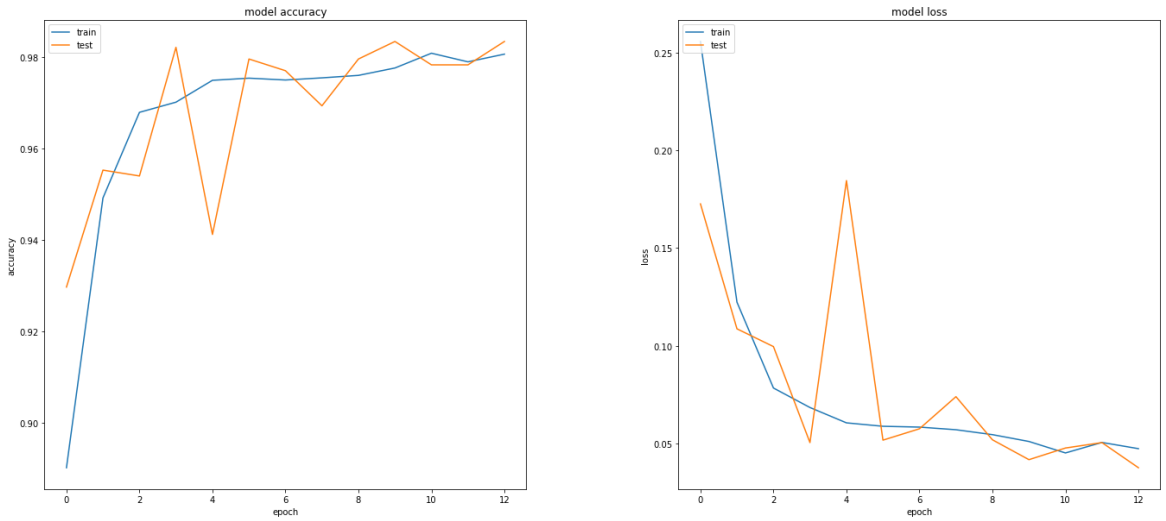


FIGURE 3.6: Predicted(_) and Actual data(_)

tual outcome (red), we can observe that in most parts of the given range, the curve in blue is totally superposed with the curve in red, hence excellent prediction which confirms the accuracy (0.9756) and precision (0.9683) shown in table 3.4, with the exception of just a small portion where the predicted value deviates a bit from the actual value.

Number of Epochs	Number of layers	Accuracy	Precision	Time of Execution(s)
13	2	0.9827	0.9452	30.6

TABLEAU 3.4: CuDNNGRU model results with 13 epochs and 2 layers



(A) Model Accuracy (B) Model Loss

FIGURE 3.7: Figures for the CuDNNGRU Model with 13 epochs and 2 layers.

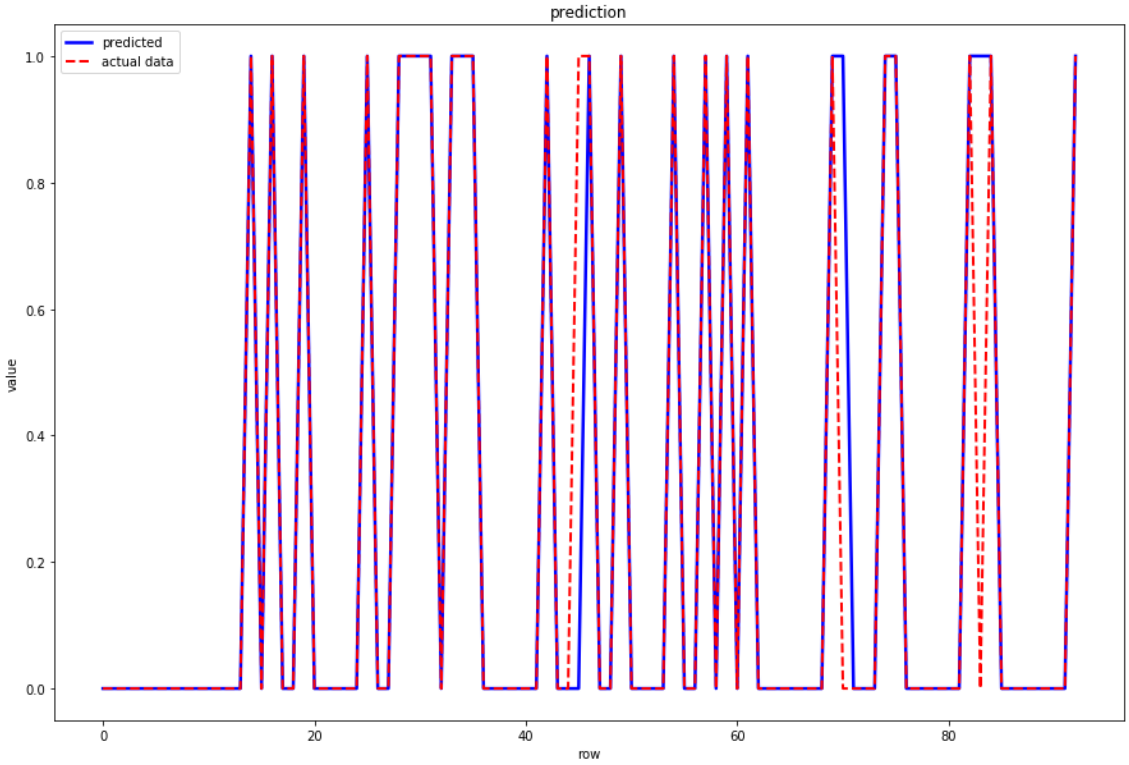


FIGURE 3.8: Predicted(_) and Actual data(_)

3.3 Simple Recurrent Network(SRN)

The results for SRN model with 2 layers and 13 epochs. Figure 3.9a shows the accuracy of the model, 2 curves are used to show this accuracy, the first being accuracy of training (blue) and the second one is the test (orange). We remark that for training, the model starts at 91% and rises to approximately 98% ,its average is approximately 94%. For test, we realize that the curve is a bit unstable starting at approximately 94% increases to 98% (max), the average approximately 95%. Training of this method is also good enough.

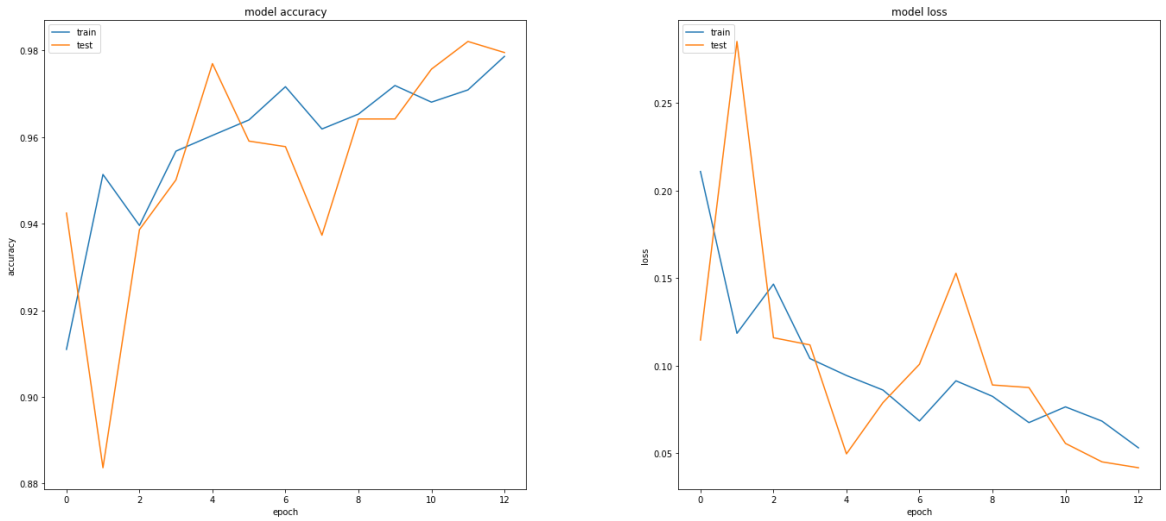
Figure 3.9b shows the model loss, for training (blue curve), the loss starts at approximately 21% and decreases up to almost 5%, on average it is between

5% and 10%. In test (orange curve), the loss starts at around 11% and there is drastic increase to approximately 30% (max), finallt decreases to about 4% having an average of around 10%, it is unstable.

Figure 3.10 is a comparison of the predicted outcome (blue) and the actual outcome (red), we can observe that in most parts of the given range, the curve in blue is totally superposed with the curve in red, hence excellent prediction which confirms the accuracy (0.9797) and precision (0.9344) shown in table 3.5, with the exception of just a small portion where the predicted value deviates a bit from the actual value.

Number of Epochs	Number of layers	Accuracy	Precision	Time of Execution(s)
13	2	0.9797	0.9344	96.9

TABLEAU 3.5: SRN model results with 13 epochs and 2 layers



(A) Model Accuracy (B) Model Loss

FIGURE 3.9: Figures for the SRN Model with 13 epochs and 2 layers.

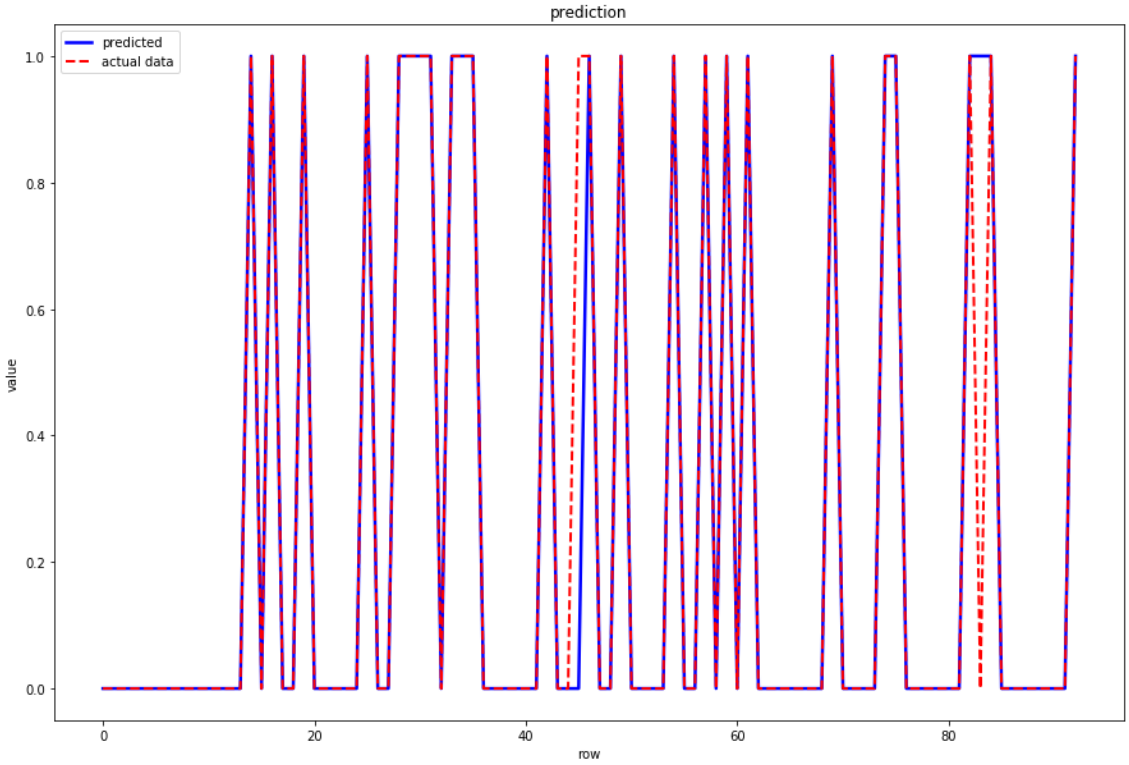


FIGURE 3.10: Predicted(_) and Actual data(_)

3.4 Discussion

3.4.1 Overall Table of Results

All the achieved results are presented in the table 3.6.

3.4.2 Interpretation

The models are executed using a personal computer with the characteristics Intel core i7-4500U(2.64GHz) CPU, 16GB RAM, and Windows operating system on the Turbofan dataset.

Tableau 3.6: Overall Table of Results

Model Used	Number of Epochs	Number of Layers	Accuracy	Precision	Time of Execution(s)
LSTM	13	2	0.9802	0.9583	162.7
CuDNNLSTM	13	2	0.9717	0.8799	33.02
GRU	13	2	0.9829	0.9433	149.58
CuDNNGRU	13	2	0.9827	0.9452	30.6
SRN	13	2	0.9797	0.9344	96.9

From the table 3.6 we can deduce that CUDA Deep Neural Network(CuDNN) makes execution of models faster, for example training an LSTM model took 162.7s, of which the CuDNNLSTM took 33s which 4 times less. In all modls, the predicted outcome given in blue and the actual outcome in red can be seen to be superposing except on small positions were there is a slight deviation from the actual expected outcome hence the models are all applicable methods for RUL estimations.

Figure **??** shows the time of execution of the 3 models LSTM, GRU and SRN. As we can see LSTM takes longer to execute as compared the others, and SRN takes the least time for execution. LSTM has three gates (input, output and forget gate) whilst GRU has two gates (reset and update gate) and SRN has no gates, hence there are fewer computations for a SRN-cell than an GRU-cell which in turn has fewer compared to LSTM. SRN trains faster though doesn't have competitive accuracy hence not the best model to use. GRU use less training parameters and therefore use less memory, execute faster and train faster than LSTM's whereas LSTM is more accurate on data-sets using longer sequence. In short, if sequence is large or accuracy is very critical, please go for LSTM whereas for less memory consumption and faster operation go for GRU. It all depends on your training time and accuracy trade off.

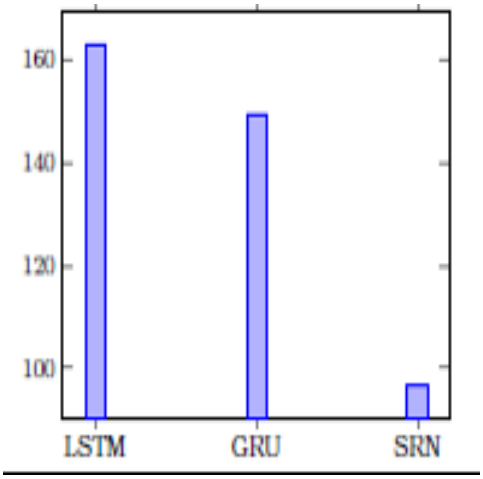


FIGURE 3.11: Graph showing time of execution of LSTM, GRU and SRN models with 13 epochs, 2 layers

Compared to LSTMs, GRU are found to be less computationally expensive due having fewer number of internal gates.LSTM and GRU outperform simple RNN in long term predictions or short term predictions.
All the models have competitive accuracy with GRU giving the highest 0.9829 and CuDNNLSTM giving the lowest. Our choice of models is then going to be based on the time of execution, as rapidity is one of the major requirements of a predictive maintenance model. It was hard to compare with the models reviewed in chapter **??** due to the fact the we used classification method and we could not calculate the RMSE which is the one used for evaluation of the best method in the papers.

Chapter 4

Future Work

4.1 Stacking LSTM cell, SRN cell and GRU cell

Cell stacking of the LSTM cell, SRN cell and GRU cell was perfomed to create models, practical the methods worked and some of them gave excellent results compared to the commonly used methods. The research was not included in this thesis because we could not get enough time to investigate its feasibility in real life application. In my future work i hope to continue with this research. The different models that were gotten after stacking are the LGR, LRG, RLG, RGL, GRL and GLR with:

- G-Gated Recurrent Neuron
- L- Long Short Term Memory
- SRN- Simple Recurrent Neuron

The figure 4.1 shows how the GLR model looks like. In our future work, we hope to implement all these methods on rel life data not simulated data.

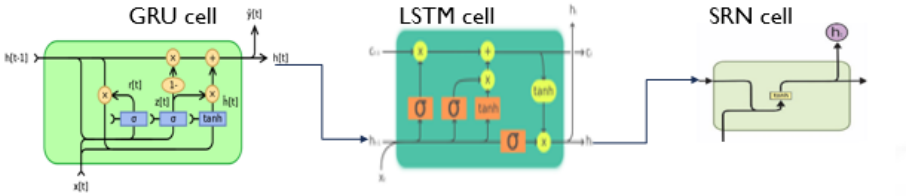


Figure 4.1: stacked cells

Chapter 5

Conclusion

In this thesis, Experiments were carried out on the popular NASA Turbofan dataset to show the effectiveness of the proposed methods. The CuDNN was seen to have great influence on the time of execution with it taking approximately 4 times less time to execute as compared to the normal none-CuDNN models. It is also important to note that the time of execution we got for all the models is less than 2 minutes, of which one can ask, why consider time of execution when they are all small (< 2 minutes)? the thing is as we increase number of layers and epochs, the model will take hours to execute and that's where the real need for CuDNN versions is seen, we couldn't do it here because of computational limitations. In a bid to investigate on which one is faster between CuDNN versions and traditional methods, we had to do the study with low epoch number and layers, then conclude that the behavior gotten is the same when we add more layers and increase number of epochs. From the experiment we can conclude that CuDNNGRU gives the best results as it has a favorable accuracy and also takes a shorter time to execute compared to the rest of the proposed methods. All other methods gave favorable results which makes them also capable of being used for RUL estimations.

Bibliography

[1] "A Comprehensive Guide To Corrective Maintenance," Apr. 2019.

[2] "Condition based maintenance strategy."

[3] "Preventative maintenance is essential to productivity," May 2019.

[4] "Predictive Maintenance (PdM): Why it Matters & How it Works," Aug. 2017.

[5] R. Amiri, H. Mehrpouyan, L. Fridman, R. K. Mallik, A. Nallanathan, and D. Matolak, "A machine learning approach for power allocation in hetnets considering qos," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2018.

[6] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mechanical Systems and Signal Processing*, vol. 107, pp. 241–265, 2018.

[7] "Components of jet engines," June 2020. Page Version ID: 962941778.

[8] C. Liu, L. Zhang, Y. Liao, C. Wu, and G. Peng, "Multiple sensors based prognostics with prediction interval optimization via echo state gaussian process," *IEEE Access*, vol. 7, pp. 112397–112409, 2019.

[9] "Understanding LSTM Networks – colah's blog."

[10] "Gated recurrent unit," Oct. 2020. Page Version ID: 983205739.

[11] "Long short-term memory," Sept. 2020. Page Version ID: 980666325.

[12] Sivaranjith, "What is maintenance? Types of Maintenance – Instrumentation and Control Engineering," Feb. 2019.

[13] S. Khan, P. Phillips, I. Jennions, and C. Hockley, "No fault found events in maintenance engineering part 1: Current trends, implications and organizational practices," *Reliability Engineering & System Safety*, vol. 123, pp. 183–195, 2014.

[14] R. Lewis and U. Olofsson, *Wheel-Rail Interface Handbook*. Elsevier, Sept. 2009. Google-Books-ID: 7QqkAgAAQBAJ.

[15] U. T. Inc, "Types of Maintenance | Compare Different Types of Maintenance."

[16] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," *arXiv preprint arXiv:1912.07383*, 2019.

[17] J. Wang, L. Zhang, L. Duan, and R. X. Gao, "A new paradigm of cloud-based predictive maintenance for intelligent manufacturing," *Journal of Intelligent Manufacturing*, vol. 28, no. 5, pp. 1125–1137, 2017.

[18] "what is a gated recurrent unit - Google Search."

[19] R. Zhao, J. Wang, R. Yan, and K. Mao, "Machine health monitoring with lstm networks," in *2016 10th International Conference on Sensing Technology (ICST)*, pp. 1–6, IEEE, 2016.

[20] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional lstm networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.

[21] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *2017 IEEE international conference on prognostics and health management (ICPHM)*, pp. 88–95, IEEE, 2017.

[22] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2017.

[23] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

[24] Y. Zhou, Y. Huang, J. Pang, and K. Wang, "Remaining useful life prediction for supercapacitor based on long short-term memory neural network," *Journal of Power Sources*, vol. 440, p. 227149, 2019.

[25] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 International Conference on Prognostics and Health Management*, pp. 1–9, Oct. 2008. ISSN: null.

[26] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis*, pp. 105–116, Springer, 1978.

[27] P. Juszczak, D. Tax, and R. P. Duin, "Feature scaling in support vector data description," in *Proc. asci*, pp. 95–102, Citeseer, 2002.

[28] M.-L. Zhang and Z.-H. Zhou, "Improve multi-instance neural networks through feature selection," *Neural processing letters*, vol. 19, no. 1, pp. 1–10, 2004.

[29] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[30] Y. Hu, A. Huber, J. Anumula, and S.-C. Liu, "Overcoming the vanishing gradient problem in plain recurrent networks," *arXiv preprint arXiv:1801.06105*, 2018.

[31] M. Nguyen, "Illustrated guide to lstm's and gru's: A step by step explanation," *Towards data*, 2018.

[32] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[33] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] J. Brownlee, "A Gentle Introduction to Long Short-Term Memory Networks by the Experts," May 2017.

[36] "7 The Simple Recurrent Network: A Simple Model that Captures the Structure in Sequences."

[37] P. Sun, W. Feng, R. Han, S. Yan, and Y. Wen, "Optimizing network performance for distributed dnn training on gpu clusters: Imagenet/alexnet training in 1.5 minutes," *arXiv preprint arXiv:1902.06855*, 2019.

[38] L. Brown, "Accelerate Machine Learning with the cuDNN Deep Neural Network Library," Sept. 2014.