University of Reading

Department of Computer Science

A practical approach to predicting remaining useful life of hard disk drives

Cristian Seceleanu

Supervisor: Prof. Richard Mitchell

A report submitted in partial fulfilment of the requirements of
the University of Reading for the degree of
Master of Science in *Data Science and Advanced Computing*

September 4, 2023

Declaration

I, Cristian Seceleanu, of the Department of Computer Science, University of Reading, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Cristian Seceleanu
September 4, 2023

Abstract

Hard disk drive failures are one of the most common causes for service interuption in data centers and as the demand for physical and cloud storage is increasing so is the likelyhood of downtime caused by these failures. Recent works reveal excellent results in predicting remaining useful life as well as classifying the current state of the drives based on S.M.A.R.T. measurements collected over relatively short periods of time however these are achieved at considerable cost in terms of complexity and resource usage combined with intricate methods for balancing the datasets which, as a whole, are fairly difficult to implement in a production environment in their current state not to mention relatively difficult to maintain and update over time. In this report a more practical approach at identifying the current state of HDDs is presented (by using the Random Forest algorithm for classification) as well as predicting their RUL (by using Bidirectional LSTM for regression) with good accuracy and confidence by eliminating the need to balance the data and by introducing an application that will ingest the measurements, generate predictions by using a hybrid approach based on multiple trained models, with minimal operational cost while keeping the model(s) up to date. The approach builds on existing works in the field, does not outperform state-of-the-art methods in terms of accuracy or confidence however it shows that the current state of HDDs can be classified with very high accuracy (greater than 90%) and that a drive's RUL can be predicted with up to 90 days in advance (not found in other works) with good levels of confidence (of over 80% R-squared)

Keywords: predicting RUL of HDDs using SMART measurements, BiLSTM on SMART data, Random Forest classification of HDD health level

Acknowledgements

I would like to express my thanks to my supervisor Prof. Richard Mitchell who has supported me throughout this research project. I would like to thank the University of Reading and its Computer Science department for their support in my academic endeavours. Last but not least I would like to thank my wife and children for putting up with me during my time in this academic programme.

Contents

List of Figures

List of Tables

List of Abbreviations

SMPCS	School of Mathematical, Physical and Computational Sciences
S.M.A.R.T.	Self-Monitoring, Analysis and Reporting Technology
SMART	Self-Monitoring, Analysis and Reporting Technology
SAS	Serial Attached SCSI
SATA	Serial AT Attachment
SCSI	Small Computer System Interface
SSD	Solid-State Drive
HDD	Hard Disk Drive
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
RNN	Recurrent Neural Network
RF	Random Forest
SVM	Support Vector Machine
DT	Decision Tree
BB	Backblaze
pRUL	predict-RUL
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
R-squared	Coefficient of determination
NAS	Network Attached Storage

Chapter 1

Introduction

This report looks at existing works in the field of identifying the current state of HDDs as well as predicting their RUL by applying machine learning on datasets consisting of SMART measurements taken at regular intervals of time and proposes an approach that is practical from a production point of view (low complexity, low cost of ownership and high return of investment) by means of a technical solution called **pRUL**.

1.1 Background

As the demand for physical and cloud storage is increasing rapidly, the number of hard disk drives in operation is also increasing and with it so is the number of disk failures. As these failures usually impact the quality of the storage services it is clear that they cannot be ignored and a more proactive approach is required (waiting for a disk to fail before replacing it is more disruptive than replacing the disk before it is about to fail).

Hard Disk Drives (or HDDs) were introduced by IBM in 1956 and since then have become the most wide-spread technology for data storage. They remain the most popular storage media in data centers even after the rise of the Solid State Drive (or SSD) – which no longer has moving parts but rather chips with storage cells - because of their price to capacity and life expectancy ratio (A. De Santo and Sperlì, 2022).

With the wide spread adoption of cloud services for workloads ranging from small (ie. individual virtual machines hosting a personal blog) to big (ie. using data science to predict or model weather patterns based on massive data sets collected over decades) it becomes clear that storage systems are required to scale to Petabytes and Exabytes which results in using hundreds of thousands and millions of HDDs per data center. At this scale disk failures are no longer rare events but rather they become the norm and with that comes the need to have optimal strategies to deal with such failures.

It is true that data loss caused by disk failure has been reduced by the adoption of solutions such as redundant arrays of inexpensive disks (RAID) however, when a disk that is part of a storage array fails and is replaced, the recovery process is a lengthy one and while it is running, additional stress is added on the remaining disks which can cause, in the best case scenario, performance degradation of the system, and, in the worst case scenario, data loss caused by the failure of one or more disks in the same storage array. This approach works however due to its reactive nature it remains an unsatisfying solution (A. De Santo and Sperlì, 2022).

In recent years focus has been shifted towards exploring more proactive solutions such as predicting when a HDD is close to failure such that the maintenance window required to replace it can be scheduled in advance to reduce the impact on the overall performance of the system (A. De Santo and Sperlì, 2022) (Lihan Hu, 2020).

Due to shifts towards predictive systems, machine learning approaches have been gaining increasing popularity - especially the ones using models trained on S.M.A.R.T. data - by relying on internal attributes of HDDs as indicators of drive health. (A. De Santo and Sperlì, 2022).

1.2 Problem statement

Two of the problems that sit at the core of this report, predicting the current state of a HDD (one) and predicting the remaining useful life of a HDD(two), have been addressed in previous papers, with a few more recent ones showing excellent results in specific contexts (and introducing state-of-the-art methods which achieve accuracy and confidence levels of over 95%) however before these approaches can be used in a real production environment one crucial aspect needs to be addressed and that is practicality (the third problem that is addressed in this report). Any business that needs to operate physical servers in order to run workloads to serve their customers must at the very least minimize the risk of data loss if not eliminate it completely and do so while balancing costs with income and with the prices presented to the end users otherwise the business becomes unsustainable. In this context a solution that requires hiring data scientists whenever a business wants to adopt a predictive approach for HDD failure, purchasing hardware for this purpose as well as collecting data from their infrastructure over long periods of time is costly and becomes unappealing early on.

1.3 Aims and objectives

The proposed research project aims to build a practical application for predicting the current state of disk drives (a first health evaluation) and their RUL using previously published state-of-the-art ML methods trained on publicly available datasets (Backblaze) which will keep itself up-to-date by continuously ingesting new S.M.A.R.T. measurements from monitored drives (never before seen HDDs) with the possible extension to SSDs and NVMEs. The first objective is to prepare a diverse enough dataset on which to train the algorithm. The data will be qualitative and sourced from public data sources afterwards it will be processed as little as possible (by eliminating invalid entries and selecting the most relevant attributes). The second objective (which is also the main one) is to identify the best technique(s) and tool(s) for handling the data and training the model (which will use either LSTM or a mix of algorithms depending on their overall accuracy) for predicting RUL. Another objective (third) is to measure the efficiency and cost of the chosen Machine Learning algorithm(s) and compare with other state-of-the-art models and techniques. The last objective (fourth) is the delivery of a practical application that can be used in a production environment to predict RUL with high accuracy in a cost effective manner and with little to no maintenance effort or operational cost. At the very least, this research project will contribute by reporting the computational and time costs of training and applying the Machine Learning algorithms on this particular type of dataset which will allow repurposing them in the future to other datasets. The main contribution of this project, if successful, will be the practical application.

1.4 Solution approach

The methodology to be adopted is a quantitative empirical experiment which will follow the Extract Clean Transform (ECT) structure: (a) Extract the data from the data sources (b) Clean the data, highlight outliers and remove them from the datasets or reweigh them (c) Create a training, testing and validation dataset (70/30/0 for classification and 35/30/35 for regression) (d) Train the algorithm(s) on the test data (e) Evaluate the algorithms and select the most effective one or a combination between them (f) Compare the selected algorithm(s) with existing results from other papers (g) Measure computational cost of each step After the training dataset has been prepared, a number of Machine Learning method(s) will be trained (with BiLSTM being the main candidate) and compared from a performance and accuracy point of view while at the same time looking at if and how they can handle new data as well as keeping themselves up-to-date training wise while ingesting streams of new data. The success of the Machine Learning algorithm will be determined by whether it is able to predict RUL with high confidence on the test dataset combined with its ability to maintain high accuracy over time (when predictions start being made by taking into account information that was not used in the initial training but rather information that the algorithm ingested over time and used to train itself) together with the operational cost required for the exercise (compute resources needed and time it takes to train and make predictions, engineering time required for operating the application).

1.5 Summary of contributions and achievements

In this work lightweight/minimal methods for data standardization, normalization, classification and RUL prediction are proposed for working with highly imabalanced S.M.A.R.T. measurements using a Random Forrest classifier to predict the current health of HDDs as well as a BiLSTM network to predict the RUL of HDDs over multiple days of look-back periods in the form of a technical solution / application called pRUL that requires minimal operational cost to provide current status and remaining useful life predictions for monitored HDDs with good accuracy and confidence which makes it a practical solution for tackling the issue at hand.

1.6 Organization of the report

This report is organised into seven chapters. Chapter 2, details the literature review of this project. The following chapter, Methodology, covers how data preprocessing and model training were performed together with the aproach taken for building the technical solution (pRUL). After that, in chapter 4, findings when experimenting with the RF and BiLSTM models are presented both on the complete dataset as well as on a subset of the dataset (experimenting on specific drive models) as well as the pRUL technical solution. Later on, in chapter 5, Discussion and Analysis, more details are presented relative to the results obtained together with their significance. In chapter 6 conclusions together with future work are covered and this report is ended with chapter 7, Reflection, where a discussion about what was learned in the process of creating this thesis is found.

Chapter 2

Literature Review

Below follows a presentation of several state-of-the-art approaches to predicting RUL either as a classification problem or as a regression problem each producing very good results.

2.1 State-of-the-art works in the field of research.

Lihan Hu (2020) proposes a model based on LSTM to predict disk failure in a given interval (30 days before the actual failure) using sliding windows and also experiments with Random Forest as a comparison (approached as a regression problem).

A. De Santo and Sperlì (2022) follows recent research in predictive maintenance, provides an overview of State-of-the-Art approaches and presents a deep learning approach to address data sparsity, need for domain knowledge and feature engineering to predict RUL of a HDD by identifying specific health conditions on the basis of S.M.A.R.T. attributes values using three main steps: defining the health degree for each HDD, extracting sequences in a specific time window for each hard disk and then assessing the health status through LSTM by associating a health level to each temporal sequence (approached as a classification problem).

A multi-instance LSTM network for failure detection of hard disk drives [3] (2020) proposes a fault prediction method based on multi-instance LSTM neural network where the data in the entire degradation process is regarded as a sample then using the LSTM network the time characteristics of the data are mined and finally a multi-instance learning method is used to treat the degradation characteristics of the full-life data as a data bag and divide it into multiple instances thus the entire life cycle data is used for HDD abnormality detection (approached as a classification problem).

A. Coursey and Sengupta (2021) proposes methods for data standardization, normalization and RUL prediction using Bidirectional LSTM network with multiple days of look- back period considering S.M.A.R.T. attributes highly correlated to failure and builds a prediction pipeline that takes into consideration the long-term temporal relations in the failure data, shows that the method is superior when compared to vanilla LSTM and establishes a baseline using Random Forest (approached as a regression problem).

In F. D. S. Lima and Machado (2018) paper the authors propose, perform and evaluate CNNs and LSTMs for RUL estimation (approached as a regression problem) and compare them with an Elman Recurrent Neural Network.

F. D. d. S. Lima and d. C. Machado (2017) proposes an LSTM model to predict RUL and compares it with Elman Recurrent Neural Network as well as a Random Forest model (approached as a classification problem). In S. Basak and Dubey (2019) a LSTM model is used to predict RUL with good accuracy (approached as a classification problem) A. Bai and Yang (2022) proposes an attention-based BiLSTM with differential features framework is proposed to further improve predicting RUL for HDDs by assigning different weights for different features at different time steps (approached as a classification problem).

Many of the works mentioned above use Random Forest as a baseline for comparing accuracy as it is acknowledged as being very good at classifying drives based on S.M.A.R.T. measurements and all attempt to predict RUL by using LSTM (either single or multilayered, vanilla, Bidirectional or attention based). In all cases the authors select one particular drive model (Seagate ST4000DM000) due to the fact that the Backblaze (2023) dataset contains a good amount of measurements for many disk drives of this particular model and perform feature selection, normalization and standardization on the dataset due to its imbalanced nature (in terms of numbers, there is very little data that indicates a bad condition vs a good condition, which is to be expected given that drives will report the same measurements over a period of days / weeks / months before a change in attribute values occurs making it very hard to predict RUL, and that the lifespan of the drives themselves is very long - usually measured in years - which results in massive amounts of data to process if one were to try to work with the complete dataset).

Some of the referenced works above, the ones that approach RUL prediction as a classification problem (for example where remaining useful life of drives is flagged as "alert" if less than 15 days before failure, "warning" if RUL is between 15 and 30 days, etc), achieve extremely high accuracy levels (90% - 95%) while others approaching it as a regression problem (where they try to accurately predict the actual RUL measured in number of days before failure) achieve very good confidence levels measured in very small R-squared, MSE and MAE values.

2.2 The project in the context of existing literature and products.

There are many products / tools available that are used for monitoring the health of HDDs by reporting the actual S.M.A.R.T. attribute values and classifying the health of the disks based on the specifications from the manufacturers and even triggering alerts when certain thresholds are exceeded however this has been shown to have an accuracy level of less than 10% (*A multi-instance LSTM network for failure detection of hard disk drives [3]*, 2020).

To address this problem many works have been published showing varying levels of accuracy when predicting the state and RUL of HDDs using various ML models trained on S.M.A.R.T. measurements however an actual product / tool that uses this approach with a high enough degree of accuracy has yet to be made publicly available for the general public be that for small or large scale environments.

A tool called "DA Drive Analyzer" (QNAP, 2023) that uses "AI" to predict failure and minimize downtime exists however it is vendor specific and targetted at NAS users thus not generally available to be used on for example personal laptops or servers in datacenters.

2.3 Review relevance analysis.

The papers mentioned above are the most relevant for this work as they cover state-of-the-art techniques for predicting the RUL of HDDs as well as to classify their current state (as "good", "bad", "fair", etc) depending on S.M.A.R.T. measurements collected regularly throughout their lifespan. They show the best results that can be obtained and in what context as well as the gaps that need filling before these approaches can become available as "simple", "practical" products / tools for anyone to use.

2.4 Critique of existing work.

While researching state-of-the-art methods that are able to predict HDD failure it was discovered that many works show excellent results when classifying HDDs as "good" and "bad"/"prefailure" or as "in good working order" or "warning" or "about to fail" as well as when predicting their remaining useful life however this is achieved based on limited diversity (usually one disk model is selected due to the quantity of data and to the severely imabalanced nature of the selected dataset not to mention due to the lack of public datasets that contain this sort of measurements).

Another find is the fact that in order to achieve high accuracy and confidence levels, careful planning and complex work is required to be performed on the datasets themselves in order for them to be used for training various prediction models which in general takes a fair amount of time and effort.

On top of the above findings, there is very little mention of keeping the models up-to-date (such that as more data is collected and the dataset evolves so should the trained models used to generate the predictions).

This report, together with the technical solution (pRUL), covers all of the above aspects by:

- experimenting with training the models on data coming from one specific model and comparing performance when predicting failure for drives of different models
- experimenting with training a prediction model using the complete dataset (trained using measurements from all drive models)
- experimenting with building ML models for each HDD model for improved accuracy (which has been mentioned and suggested in previous publications in the field)
- collecting information from the HDDs that are being monitored by the pRUL agent, sending the measurements to the main server, storing and using the measurements to update the ML model at the right time and then generating predictions using the updated model
- by creating a product / tool that anyone can use, the solution can collect S.M.A.R.T. measurements from a much larger pool of devices (with owner's approval) thus making the dataset more robust which will also allow for training better prediction models and extend to other types of disk drives (SSDs and NVMEs)

2.5 Summary

In this chapter the review of the state-of-the-art literature in the field was covered, a description of how the review is relevant to this report was presented, gaps in the existing literature were identified and a listing of how this work will fill them was provided.

Chapter 3

Methodology

Given that the aim of this report is to predict the remaining useful life of hard drives used in both small and large scale environments (from a personal laptop to a cloud service provider) and do so in a practical way, the approach has to take into account existing works in the field, improvements that can be made without having to reinvent the wheel, deliver on the promise in a cost effective and timely manner with the highest precision and confidence possible while at the same time be as easy to use and operate / maintain as possible. With this in mind the idea to create a technical solution (called pRUL) came to be that would have 3 main components:

- (agent) an agent application that is to be installed and configured on the systems where the HDDs are in use (that will support a wide range of operating systems),
- (webapi) a web API service that acts as a collector for S.M.A.R.T. measurements sent by the agent as well as an oracle that provides predictions of the state and RUL of the disk drives when requested by the agent application and
- (oracle) an update service that is used to keep the prediction models up-to-date by constantly training them on new data as it becomes available. Depending on the size of the infrastructure, the users may decide to deploy the complete suite in their own infrastructure (in which case they become responsible for operating and maintaining the technical solution) or decide to install only the agent on the nodes they want monitored (in which case they will rely on a third party for the operational and maintenance tasks associated with the platform).

Considering that when deploying the pRUL solution there is no historical data for the HDDs the users want to monitor, an argument can be made that a two step approach is needed as follows:

1. an initial assessment of the health condition is performed using a classification model based on Random Forest that will report the state as good (if a drive have more than a set threshold of 30 or 60 or 90 days before failure) or prefail if (the drive has fewer days remaining than the set threshold before failure) after which
2. as measurements start to be collected on a daily basis and once enough measurements have been collected for each of the disks that are being monitored a regression model based on BiLSTM will be used to predict the actual number of days that the drive has before failing.

Due to time constraints, the first version of the application is a minimum viable product / proof of concept at best so it focuses on the immediate need which is to flag disks which look like they are about to fail in the next 14 / 30 / 60 / 90 days and estimate with a good level of confidence how many days are left before the actual failure however, as you will see later on in this report, future versions of the technical solution will have the ability to assess the health level relative to more classes as well as predict the RUL with more time in advance which will give an even longer window to plan the maintenance (required to replace the drives that are going to fail);

In the next sections a description in more detail of each component of the pRUL solution follows.

3.0.1 Data Preprocessing

In what follows a description of how data preprocessing was approached is presented together with some information about how S.M.A.R.T. works followed by a few details about the dataset selected for this work.

S.M.A.R.T.

S.M.A.R.T. stands for Self-Monitoring, Analysis, and Reporting Technology and is a technology found in disk drives (HDDs, SSDs and NVME SSDs). It is independed from the Operating System, BIOS, or other hardware as it is built into the drives themselves. SMART was invented because something was needed that could monitor the health state of disk drives and its purpose is to report if a drive is about to fail.

The Backblaze dataset

Backblaze (2023) is a company that provides Cloud Storage. The company was founded in 2007 and since 2013 it started collecting daily S.M.A.R.T. measurements from each of the drives used in their datacenters and releasing quartely archives (containing CSV files with daily measurements) on their website. Together with the data they also provide statistics and insights based on the hard drives in their datacenters (Backblaze, 2016).

The Backblaze dataset (up to and including q4 2022) contains daily measurements collected from 335282 hard disks. Each entry consists of information about the hard drive (model, serial number, capacity in bytes) together with raw and normalized values for 90 SMART attributes and an added feature called *failure* which if set to 0 it represents that the drive is working and if set to 1 it represents that either the drive has failed (or in some cases is about to fail as it is causing issues) and has been removed by Backblaze technicians from the live environment.

Data Collection

At first a MySQL database was created with one table (called drive_stats) with the intention of importing programmatically all the Backblaze data into it which was successfull however, the resulting database (with a size of apx 240G and containing over 350 million entries ... 367203010 to be exact) was slow and impractical to work with so a decision was made to split it into multiple smaller databases (one for each quarter of every year resulting in 40 databases named using the convention backblaze_year_quarter ex. backblaze_2019_Q1).

Upon an initial inspection of the data a discovery was made and that is that the dataset contained measurements for 90 attributes (which is justified by the fact that S.M.A.R.T. attributes are vendor and model specific - also not all attributes apply to all hard drives, some apply to SSDs others apply to HDDs) and given that the goal set in this report is to build a model that predicts failure foing forward all work will be done relative to drives which have failed (the *failure* attribute is set to 1 at some point in time) so a new database was created (called backblaze_ml_full again with one table named drive_stats) where all data collected from the drives which have failed was imported programmatically.

This had the effect of reducing the total number of entries in the database to close to 15.5 million entries (15653251 to be precise, corresponding to 17732 HDDs which have failed over time) and the size on disk of the database to apx 8G which made it practical to work with.

Feature Selection

The feature selection process was performed over a number of steps which resulted in the creation of a separate database called backblaze_ml (again containing only one table called drive_stats) holding only the data found to be relevant for this work.

At first all columns were removed that corresponded to SMART attributes which had no values collected after which analysis was performed to identify which of the attributes are indicators of failure and which are indicators of performance or simply counters (for example attribute 195 - Hardware ECC Recovered - was flagged as an indicator of wear/performance as it simply counting the number of errors that were corrected - the higher the number the lower the performance of the drive however the drive is still operational, similar for 194 - Temperature Celsius - as in a controlled environment such as a datacenter the temperature will fluctuate very little, similar for 183 - SATA Downshifts - which is actually an indicator of problems with the SATA cable rather than with the drive itself) and eliminate all that were identified as not relevant for failure. After this step the resulting database contained measurements for the following SMART attributes:

- 5 - Reallocated Sector Count (BB)
- 187 - Reported Uncerrectable Errors (BB)
- 188 - Command Timeout (BB)
- 189 - High Fly Writes
- 196 - Reallocation Event Count
- 197 - Current Pending Sector Count (BB)
- 198 - Uncorrectable Sector Count (BB)

noting that 5 of the attributes were also selected by Backblaze as indicators of failure during their analysis (Backblaze, 2016).

After applying the Pearson correlation between the selected attributes a strong correlation was identified between 197 and 198 and, unlike Backblaze (who decided to keep both for their model), for this report, 197 was kept due to the fact that more drives report it and that there are more measurements for it present in the dataset.

Another interesting fact discovered during this stage was that some entries in the dataset came from SSDs which unfortunately had to be filtered out as they are not in scope for this report, not to mention they are not enough to use as a dataset for traininig a separate model for SSDs).

Closer inspection of the dataset revealed that the it contained raw measurements collected for 6 attributes (out of the 255 possible) up to March 2014 after which more started being

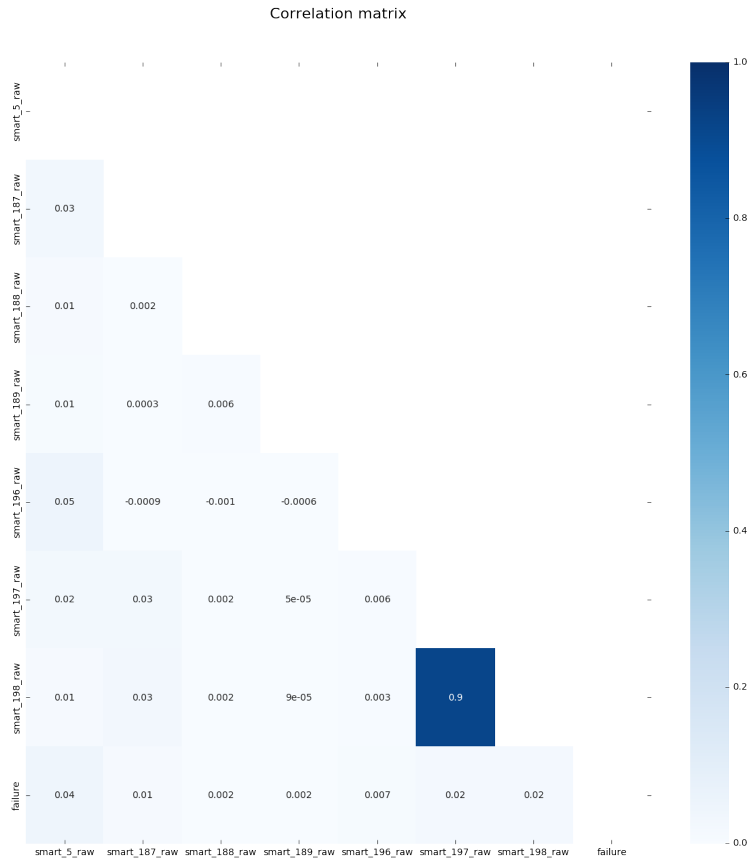

Figure 3.1: Pearson Correlation.

collected (24 features for which both raw and normalized values started being reported) and in this context, given the SMART attribute selection above, measurements collected before said date were ignored.

One last thing to note here is that the SMART attributes were chosen based on their relationship with failure (how relevant they are in identifying if a disk should be marked as good or in a pre-fail/failed state) and not taking into account their evolution over time. In other words the feature selection work presented so far was done in the context of the first objective - the initial health evaluation of a HDD.

For the second objective - where the aim is to predict the RUL of the HDDs (the number of days before the actual failure) - when experimenting with the selected attributes it was observed that they are not good indicators of failure over time - in other words, when experimented with training several regression models using the selection of attributes mentioned above, the trained models did not produce good results. After revisiting the literature it was decided to select the same SMART attributes as A. De Santo and Sperlì (2022):

- 1 - Read Error Rate
- 3 - SpinUp Time
- 5 - Reallocated Sector Count (BB)
- 7 - Seek Error Rate
- 9 - Power-On Hours

- 187 - Reported Uncerrectable Errors (BB)
- 189 - High Fly Writes
- 194 - Temperature Celsius
- 197 - Current Pending Sector Count (BB)

(and used their normalized values for training the regression models as will be seeb later on) because they are better indicators of wear over time and lead to improved performance of the regression models.

Data Standardization

For the first objective no standardization method was used. After cleaning the data (replacing missing/Nan values with 0) the raw data was fed to the classification model based on Random Forest (as it can handle outliers) and work with it without further transformations not to mention from a practical point of view it keeps things simple. For the second objective MinMaxScaler was used on the normalized values of the selected SMART attributes to scale them to the same interval [-1, 1] as this improves the time it takes to train the model.

3.0.2 Model Training

In this section Random Forest, LSTM and BiLSTM are introduced and aspects regarding training the models as well as testing and validation of the models are covered.

Random Forest

Random Forest is a commonly-used machine learning algorithm which combines the output of multiple decision trees to reach a result. RF algorithms have three main hyperparameters which need to be set before training: node size, the number of trees and the number of features sampled. From there the random forest classifier can be used to solve regression and classification problems. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.

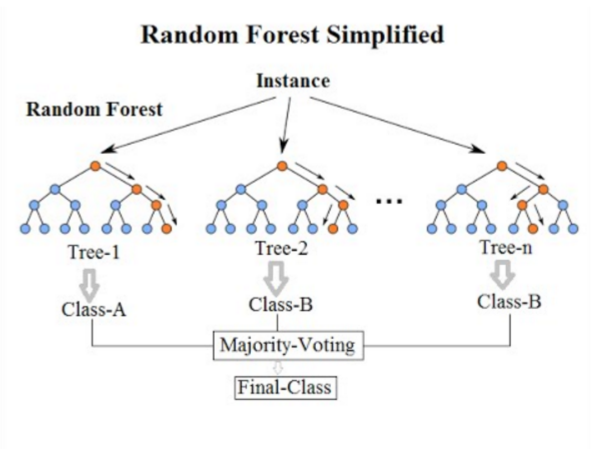

Figure 3.2: Random Forest (Wikipedia).

Health Assessment through Random Forest

As mentioned before, the first objective of this report is to provide an initial health assessment of a HDD by creating a classification model based on the Random Forest algorithm that looks at the state of a drive at a point in time (the raw values of the six SMART attributes of interest) and classifies its state as either good or prefailure. To achieve this, a model was trained on the dataset containing measurements from all the drives that have failed as follows.

Using the failure attribute present in the dataset together with the date of each measurement for each drive a new attribute called *prefailure* was created which takes either 0 or 1 as a value indicating if a drive is in good (0) or prefailure (1) state. The value of the prefailure attribute is set to 1 for all measurements collected within the last 14 / 30 / 60 / 90 days before the actual failure (the last entry for each drive, when the drive has failed has been removed from the dataset as the aim is to give the user a reasonable window of time for the maintenance to occur not inform the user that the drive has failed). The dataset is then split into 70% training and 30% testing data and the RF models are trained. After training the model its accuracy, precision, recall and f1 scores are measured against the test data and compared. At first experiments were performed using this model on the data collected for a specific HDD model (Seagate 10TB ST10000NM0086) afterwards the model was trained on the whole dataset.

LSTM and BiLSTM

Long short-term memory (LSTM) network is a recurrent neural network (RNN), aimed at dealing with the vanishing gradient problem present in traditional RNNs. LSTM introduces a cell state that contains a series of gates in order to gain more control over the information that is retained between cells. LSTM consists of three gates: forget, input and output. The combination of these allows the LSTM cells to extend thir short term memory, keeping any information needed to go through the entirety of the learning process. Each gate contains neural networks that serve a specific purpose and contain activation functions such as *sigmoid*. The forget gate takes in information from the previous cell and current input to decide what to keep or forget. Whatever information is kept goes through the input gate. This determines what values will be updated in the cell. The *tanh* function is applied on the cell state and current input for regulation. The cell state is then updated according to the combination of forget and input gates. Using the current cell gates and state, the output gate decides what to pass on to the next cell. A diagram outlining the LSTM cell is shown in Fig. 3.3

Numerous variants of LSTMs have been introduced to improve performance one of which is the Bidirectional LSTM or BiLSTM that is used in this report. A bidirectional LSTM is a variant of the LSTM that consists of two LSTM layers which run at the same time. One runs on the input sequence in the forward direction and the other runs backwards on the input sequence. This way LSTM runs in both directions. In the context of this thesis one could think of one direction of the LSTM running on the sequence of hard drive data leading up to failure and another running on the sequence backwards from failure. This allows the LSTM to better learn the relationship between the features and the remaining useful life with a simple, low-cost architecture change which is the reason behind choosing BiLSTM to determine the RUL of HDDs.

Remaining Useful Life through BiLSTM

The second objective of this report is to predict the actual number of days before failure by creating a regression model based on the BiLSTM algorithm that looks at a series of

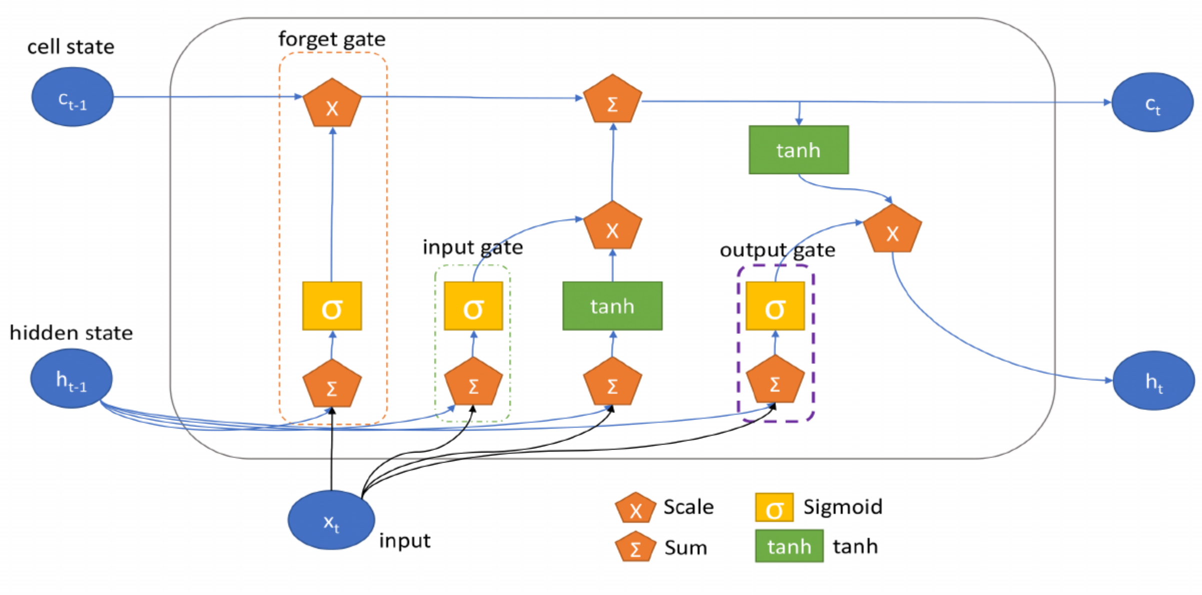

Figure 3.3: Structure of LSTM cell.

measurements (the normalized values of the nine SMART attributes of interest) over a number of days (14 / 30 / 60 / 90) before the actual failure.

To achive this, a model using three BiLSTM layers (the first with 128 units, the second with 64 units and the third with 32 units) followed by three Dense layers (the first with 96 units, the second with 128 units and the third with 1 unit) was trained on the dataset as follows:

- at step1 MinMaxScaler was applied on the normalized values (to scale all values to interval [-1, 1])
- at step2 the dataset was transformed such that for each individual disk at least 30 consecutive entries of 14 / 30 / 60 / 90 day lookback windows exist together with the actual RUL
- at step3 the dataset is split into 70% train and 30% test with the train data further split 50/50 into train/validate resulting in an actual split of the original dataset of 35% train, 35% validate and 30% test
- at step4 the model is trained on the train set in batches of 500 using MSE as the loss function to check against the validate set over 150 epochs - this step is repeated several times until the loss function shows signs of overfitting or signs that the model is not learning anymore
- and finally at step5 the model is trained on the validate set in batches of 500 over 25 epochs

Because this is a regression model its performance is measured using MAE, MSE, RMSE and R-squared as indicators. As before, at first experiments were performed by using this model on the data collected for a specific HDD model (Seagate 4TB ST4000DM000) afterwards the model was trained on the whole dataset. As a comparison this model was also trained on the exact same dataset used by A. De Santo and Sperlì (2022), which is publicly available, as well as on a variation of the dataset based on the local database generated without applying the balacing step (which was to replicate the sequences belonging to the minority classes), used in the reference paper, and compared results.

3.0.3 pRUL

In this section the technical solution used as the practical application of this report - called pRUL - is covered which is the last objective as mentioned in Section 1.3 together with its three main components.

The agent

The pRUL agent is a minimal application that can run in a variety of forms (such as a desktop application, a background service as well as a cron/task scheduler job). Its purpose is to collect information from the target machine about its HDDs (as specified in its configuration file) and send it to the web API service and retrieve predictions with regards to the state and RUL of each of the monitored HDDs. In theory the agent could generate predictions locally provided it comes with the trained models however this is impractical at scale as it would use too many resources on the target machine.

At the time of writing this thesis the agent comes in one flavor: a python script together with a configuration file that can either be executed manually or as a cron job on Unix based systems. To configure and use the agent the user must first specify the authentication details then the path to the smartctl command (if not standard) together with the path to the disk drive (and the driver if using a RAID controller such that smartctl can retrieve the SMART data) after that it is up to the user to configure it as a daily cron job or manually run the script every day.

The web API

The pRUL web API is, at its core, a collector for S.M.A.R.T. measurements in that once received it stores them in the database. Upon receiving the measurements it then relies on the oracle service to provide predictions which once available it relays to the agent. For practical reasons the API is written in PHP and uses a MySQL database to store the data and has a minimal set of requirements with respect to libraries and hardware resources (it will work on any shared hosting provider, container or entry level virtual machine). Due to its design it can scale from running on a single machine (for small deployments) to a cluster of machines (for large deployments).

The oracle update engine

The pRUL update and prediction service, also called the oracle, is used to keep the prediction models up-to-date by constantly training them on new data as it becomes available as well as to generate predictions on the data collected by the agent. The predictions are saved in the database so that the API service can access and send them to the agents.

For practical reasons the oracle service is written in python (because of its diversity of machine learning libraries) and connects to the same database as the API service to load the SMART data which it needs to generate predictions.

Due to its design it can scale from running on a single machine (for small deployments) to a cluster of machines (for large deployments) and requires a fair amount of resources for generating predictions (a minimum of 8 CPU cores and 32G RAM is recommended).

As is expected, for updating the models, depending on the size of the dataset, a minimum of 16 CPU cores and 128G RAM is recommended (this is the hardware specification used during the development phase that worked best). As soon as a new model is available it is immediately used to generate predictions (no human intervention is required).

As stated before, the oracle service uses two machine learning models for generating predictions - the first one is used to identify if a drive is prefail state (with failure predicted in the next 90/60/30/15 days) and if so the second one is used to predict the number of days left before the actual failure (RUL). Unfortunately RF does not support incremental training meaning that keeping the classification model up-to-date requires retraining the model on the whole dataset every so often (when its accuracy drops below a specific threshold) however BiLSTM does so in this context whenever enough new data becomes available the model can be updated by fitting the existing model on the new data.

At the time of writing this thesis the oracle service uses the CPU for training the models however in the future it will be extended to use GPUs as alternative if available.

3.1 Summary

In this chapter detailed information was provided about the approach to predicting the health state and RUL of HDDs, what this report means by *practical*, how data preprocessing, data collection, feature selection and data standardization were approached, a description of what SMART for HDDs is, an introduction of the machine learning algorithms used for the models presented in this report was provided together with their purpose in context of this work and the high level specifications of the pRUL technical solution, with its three components (agent, web api, oracle), were presented.

Chapter 4

Results

In this thesis the aim is to assess the current health state of HDDs and predict their RUL by using state-of-the-art approaches and do so in a practical way then discuss and compare results with other works in the field. Random Forest was chosen for training a classification model that would provide the initial health evaluation of a HDD and BiLSTM (a flavor of LSTM) for training a regression model that would predict the RUL of a HDD and with these models a technical solution called pRUL was built that can be used in live/production environments for predicting disk failures.

4.1 Health Assessment through Random Forest

As described in Section 3.0.2, the RF model was trained to predict the prefailure state of HDDs with 14 / 30 / 60 / 90 days before actual failure on the complete dataset and experiments were performend with training the model on a specific disk model and checked its performance against a related but different disk model.

Table 4.1: Random Forest model predicting prefailure state

Data	Days	Accuracy	Precision	Recall	F1	Notes
All	14	0.986	0.795	0.193	0.311	-
ST10000NM0086	14	0.953	0.142	0.159	0.150	train 100% self, acc on ST12000NM
All	30	0.971	0.835	0.209	0.335	-
ST10000NM0086	30	0.996	0.877	0.716	0.787	train/test 80/20, acc on self
ST10000NM0086	30	0.953	0.142	0.159	0.150	train/test 80/20, acc on ST12000NM
All	60	0.944	0.861	0.213	0.341	-
All	90	0.918	0.851	0.220	0.350	-

As can be seen in Table 4.1, the best accuracy (98.6%) was obtained when trained to identify prefailure at 14 days before actual failure (with a precision of 79.5%) however, the best precision (86.1%) was obtained when training the model to identify prefailure at 60 days before failure (with an accuracy of 94.4%).

A few experiments we also performed to determine if a model cound be trained on measurements collected from a specific HDD model and then use that model to predict failure for similar drives and in terms of accuracy the model performed well however in terms of precision it performed poorly (¡15%).

4.2 Remaining Useful Life through BiLSTM - wip

The BiLSTM model, as mentioned in Section 3.0.2, was trained to predict RUL using 30 x 14 / 30 / 60 / 90 day lookback windows and as can be seen in table 4.2 the best results were obtained when training on 30 x 90 day lookback windows for the complete dataset and on 30 x 60 day lookback windows for the ST4000DM000 dataset.

Table 4.2: BiLSTM model predicting RUL

Data	Windows	MAE	MSE	RMSE	R-Squared
All	30 x 14	3.941	29.394	5.422	0.655
ST4000DM000	30 x 14	4.714	40.971	6.401	0.519
All	30 x 30	-	-	-	-
ST4000DM000	30 x 30	-	-	-	-
All	30 x 60	3.563	24.018	4.901	0.718
ST4000DM000	30 x 60	2.431	12.290	3.506	0.856
All	30 x 90	2.947	16.863	4.106	0.802
ST4000DM000	30 x 90	2.772	13.994	3.74	0.836

An experiment was also performed where the model presented in this report was trained on the dataset used by A. De Santo and Sperlì (2022) and it results were recorded in table 4.3.

Table 4.3: BiLSTM model on A. De Santo and Sperlì (2022) dataset to predict RUL

Windows	MAE	MSE	RMSE	R-Squared
30 x 14	0.915	4.929	2.220	0.938

4.3 pRUL

From a practical point of view, the problem of identifying the current health state of a HDD together with predicting when it will fail is difficult to approach. To address it the technical solution - called pRUL - was created which solves it by using the two machine learning models described in this report as follows.

When a measurement is received from the agent it is first fed to the RF model which will identify if the drive is in a prefail state at 90 / 60 / 30 / 14 days (will check against all) and if the HDD has a positive result in one or more of the four time frames then the worst case scenario prediction is returned to the pRUL agent to inform the user (if for example prefailure is flagged at 60 and 30 days then the user will be informed that a drive will fail in the next 30 days rather than in the next 60). If the drive has been monitored for long enough (in other words, if enough measurements have been collected such that the oracle service can build a lookback window of 14 / 30 / 60 / 90 days) matching the corresponding windows flagged by the RF model they will then be fed to the BiLSTM model to generate predictions specific to each matching lookback window which will be sent to the agent to then report back to the user. Building on the previous example, once at least 30 days worth of measurements have been collected BiLSTM can then be used to predict RUL, once 60 days worth of measurements have been collected then predictions cab be issued for both 30 and 60 day lookback windows and again return the worst case scenario value to the user - if for example the 30 day lookback model returns 28 and the 60 day lookback model returns 26 then the application return 26 days as the RUL to the user.

Algorithm 1 illustrates the logic described above which, as can be seen, is fairly simple (mostly if statements).

In this way a notification can be sent to the user that a disk is about to fail in the next 14 / 30 / 60 / 90 days together with an approximation of the actual number of days the disk has left before failure. The user will then be able to act on this information by replacing the disk drive at a suitable time or disregard the warning and wait until the drive actually fails before replacing it. Either way, the user is informed about the state of the HDD and can take measures to mitigate the impact of the approaching disk failure by for example making sure there are enough spares in stock.

4.4 Summary

I this chapter the results that were obtained by applying the methodology described in the previous chapter when training the two models (Random Forest for classification and ByLSTM for regression) on both the complete dataset as well as on a subset of it that contains drives of a particular model (ST10000NM0086 in RF case and ST4000DM000 in BiLSTM case) were presented together with the pRUL technical solution which is the result of creating a software application that aims to use the two ML models to classify the state of HDDs and predict their RUL in days.

Algorithm 1 Algorithm used by the oracle service to return health state and RUL of HDDs

Input: $[\mathbf{x}, \ldots, \mathbf{z}] = [(x_1, x_2, \ldots, x_N), \ldots, (z_1, z_2, \ldots, z_N)]$ ▷ takes multiple sets of measurements over time

Output: $prefailure$, rul ▷ returns the health state and rul

```
function PREDICT([x, ..., z])
    prefailure ← 0                                   ▷ set prefailure to 0 meaning drive is good
    rul ← NULL                                       ▷ no prediction for RUL
    healthState ← identifyState([x, ..., z])
    if healthState[failure_in_90] == 1 then
        prefailure ← 90
        predictedrul ← predict_rul_90([x, ..., z])   ▷ model is called to predict rul with 90 days lookback windows
        if ((predictedrul! = NULL)and(rul > predictedrul))or(rul == NULL) then
            rul ← predictedrul                       ▷ set rul to the predicted value
        end if
    end if
    ...                                              ▷ repeat the if statments for 60 and 30 day predictions
    if healthState[failure_in_14] == 1 then
        prefailure ← 14
        predictedrul ← predict_rul_14([x, ..., z])   ▷ model is called to predict rul with 14 days lookback windows
        if ((predictedrul! = NULL)and(rul > predictedrul))or(rul == NULL) then
            rul ← predictedrul
        end if
    end if
    return (prefailure, rul)
end function
function IDENTIFYSTATE([x, ..., z])
    prefailure ← {failure_in_14 : 0, failure_in_30 : 0, failure_in_60 : 0, failure_in_90 : 0}
    if predict_state_90([x, ..., z]) == 1 then       ▷ model is called to predict prefailure within 90 days
        prefailure[failure_in_90] ← 1
    end if
    ...                                              ▷ repeat the if statments for 60 and 30 day predictions
    if predict_state_14([x, ..., z]) == 1 then       ▷ model is called to predict prefailure within 14 days
        prefailure[failure_in_14] ← 1
    end if
    return prefailure
end function
```

Chapter 5

Discussion and Analysis

In this chapter an evaluation and analysis will be provided based on the results presented above their implications together with several conclusions will be brought forward.

5.1 Deep dive into the results

5.1.1 Random Forest model

Looking at the results obtained for the classification model based on the Random Forest algorithm, table 4.1, an observation can be made that the best accuracy is obtained when training the model to predict the prefailure state at 14 days before actual failure and that the best precision is obtained when training the model to identify prefailure at 60 days before failure with the second best precision value being obtained at 90 days prefailure. An interesting observation is that the accuracy of the model is above 90% in all cases and that it seems to go down the further away from failure the prefailure state is marked. Based on these observation a statement can be made that the ideal number of days before failure used to indicate prefailure for a HDD should be close to if not equal to 60 days if the goal is to have both high performance and accuracy.

The low values for the recall and F1 scores are the effect of the dataset being highly imbalanced (as it contains mostly entries indicating a good HDD state - years worth of daily measurements - and a few entries indicating a drive in prefailure state - coresponding to the number of days before failure flagged as the prefailure state used in training the model) and of the decision to not use any rebalancing techniques on it due to practical reasons mentioned before in this report.

The extra experiments performed with training the RF model on a particular HDD model and then using it to predict prefailure state on a similar/related but different model produced very good results with respect to accuracy but in terms of precision they performed poorly. As can be seen, these extra experiments confirm a suspicion which is that the best way to classify the health of a drive is to train a machine learning model using disks of same make and model because SMART attributes are vendor and disk specific (some are common across multiple models from same manufacturer and sometimes even across multiple manufacturers but in general they are not, and even when they are common across model and/or manufactures they report values measured using different units or differently altogether).

Another interesting observation can be made when comparing results with the RF model presented by A. De Santo and Sperlì (2022) (table 5.1) where the accuracy of the model presented in this paper is considerably higher than the one from the referenced paper and the reason behind this is detailed in the feature selection section of this thesis (for this report, the selection of the features was done based on which attributes are better at indicating failure in context of classification as opposed to the reference paper where the features were selected based on their ability to indicate degradation over time leading to failure).

Table 5.1: Model comparison with A. De Santo and Sperlì (2022) on BB dataset

Model	Accuracy
Proposed RF model	0.986
Referenced RF model	0.858

More experimentation was performed with training a regression model using the Random Forest algorithm on the dataset presented in this report (both on the complete version as well as on a subset containing measurements from specific HDDs of a particular model) and the results that were obtained were poor and deemed as not relevant for this report as this was already confirmed in previous works in the field (the initial premise was that it was worth a try given that previous experiments were performed on smaller datasets specific to one disk model and maybe if feeding the model more data from more disk models over a longer period of time would make a difference, it did not).

5.1.2 BiLSTM model

In the case of the BiLSTM model, Table 4.2, the most interesting observation that can be made is that the model trained on the complete dataset performs best with longer lookback windows (90 days) which again confirms an initial suspicion that the longer the lookback window the better the RUL prediction considering that disk drives take a long time to degrade and ultimately fail.

Another interesting observation that needs to be pointed out is that, as per table 4.3, the proposed model performed very well (with MAE ¡ 1 and R-squared of 93.8%) on the exact same dataset used in the reference paper (A. De Santo and Sperlì, 2022) which means that, if it was possible to replicate their rebalancing step in a practical (easy) way at a later point in time, better results can be expected when training the model presented in this thesis on the complete dataset. In the referenced paper, the authors used LSTM to predict RUL by classifying the health level of HDDs into a number of different classes such as Good, Very Fair, ... , Alert and achieved excellent results thanks to the rebalancing operations performed on the dataset containing measurements from a single HDD model - ST4000DM000.

The findings were also compared with the work of A. Coursey and Sengupta (2021) (5.2) where they predicted RUL using a regression model trained on the last 60 / 120 days before failure using lookback windows of 5 / 10 / 15 / 30 and where they marked actual RUL greater than 30 as being 30 for training the model (in other words, a drive which has more than 30 days before failure is a drive in good working order) and where they scaled each feature from each drive as having a mean of 0 and a variance of 1 (and in doing so lost the ability to apply the same scaler to the test set since each scale was dependent on the hard drive). Another experiment was performed where the model presented in this report was trained with more than 30 lookback windows (45 x 15 day lookback, 60 x 15 day lookback, 90 x 30 day lookback, 120 x 30 day lookback) but due to the poor performance of the model the decision was made to not include the results in the thesis. A similar result can be observed in table 5.2 for 105 x 15 and 90 x 30 where R-squared dropped considerably compared to 45 x 15 and 30 x 30 which indicates that training the regression model using too many lookback windows does not perform better.

Table 5.2: BiLSTM comparison with A. Coursey and Sengupta (2021)

Model	Windows	MAE	MSE	RMSE	R-Squared
Proposed model	30 x 15	3.941	29.394	5.422	0.655
Referenced model	45 x 15	0.120	-	-	0.998
Referenced model	105 x 15	4.874	-	-	0.071
Referenced model	30 x 30	0.132	-	-	0.998
Referenced model	90 x 30	6.792	-	-	-0.565

5.2 pRUL

Initially, when setting out to address the problem of predicting the RUL of HDDs, the idea was to look at measurements collected over the entire lifespan of HDDs and try to predict when they will fail however after doing research on the topic it was revealed that this is not feasable at this point in time (the amount of resources and the time available to deliver a solution were not sufficient) so this problem was approached from a different angle by looking at other, more realistic, ways of achieving the same goal. By combining the results of the two models (RF and BiLSTM), the oracle service is able to catalogue drives into good and prefailure at different timeframes before failure (90 / 60 / 30 / 14 days) and relative to these can then generate predictions that are more specific with respect to the actual remaining number of days before failure.

5.3 Significance of the findings

Based on the results shown in this thesis the following statement can be made: at this point in time, using the state-of-the-art algorithms and methods, the health state of a HDD can be determined with very good accuracy and precision (with prefailure state marked at 90 or 60 or 30 or 14 days) and also a prediction for its RUL, about how many days are left before the drive will fail and needs to be replaced, can be provided with a high confidence level and not only that but this can be done in a practical way thanks to the pRUL technical solution.

5.4 Limitations

As is the case for any technical solution, pRUL has certain limitations, some more obvious than others. One such limitation is that it relies on S.M.A.R.T. measurements collected over time (more specifically at regular intervals of time) which means that it can only be used to predict failure for disks which have this functionality (it cannot predict failure for other storage media such as USB sticks or DVDs).

Another important limitation is that at this point in time it cannot predict failure for SSDs or NVMEs due to the fact that there are no publicly available datasets containing measurements from these types of drives. An improvement can be made at a later point in time thanks to its ability to collect S.M.A.R.T. measurements in that a model can be built after collecting enough data from agents deployed on systems that use SSDs provided that the owners of the systems are willing to share these measurements even though at first they will not benefit from any useful prediction.

Also, due to the *practical* nature of the project, there are limitations with respect to its accuracy, precision and confidence level caused by the severely imbalanced nature of the dataset which, in theory, can be overcome to some extent by using some balancing techniques on the dataset and potentially better ML algorithms or more experimentation, however, this will come with a high price tag.

Throughout this report it is mentioned that the solution is *scalable* due to its design, which is true, however, should the solution be deployed and used at a true global scale (for example to monitor every HDD and/or SSD on the planet), it would suffer from limitations such as network bandwidth and latency, availability of compute and storage resources (CPU cores, RAM, disk space to store all the measurements) and training ML models on massive amounts of data. This limitation can be overcome by deploying multiple regional independent instances, each addressing the problem of predicting failure for a portion of the disks which would also come with a high price tag but would work as this is a model that cloud service providers use to provide services at global scale (regional functionality).

And, as a closing note, possibly the most important limitation comes from the fact that the data needed to train the models that predict failure, must come from disks which have actually failed. Given that the average lifespan of disk drives is measured in years and that every year new models are released with more storage capacity there is also a chance that this *practical* solution becomes obsolete over time as it will only be able to predict failure for older models of disks which nobody is still making or using. At best it could end up being practical only for environments where storage resilience is more important than increased capacity or increased performance (which usually come with newer disk models). In general, before hard disks are made available to the general public they are tested by disk manufacturers in their facilities and afterwards in production environments with the help of some of their bigger customers and in this context it is possible to mitigate this limitation by using the proposed technical solution to collect measurements at this stage such that once new disk models become available to the general public pRUL can reliably provide predictions to its users.

5.5 Summary

In this chapter a closer look was taken as the results, they were compared with other works in the field, limitations of the solution were identified and potential improvements that can be made and under which circumstances were suggested.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis the state-of-the-art works in the field of evaluating the health level of HDDs and predicting their remaining useful life using SMART measurements collected over time at regular intervals were studied and the conclusion that the existing approaches have not yet produced a practical solution that can be made generally availalbe to consumers was made, a gap which was filled by creating one such practical application (called pRUL), which is also the greatest contribution this report brings to this field, that uses a combination of ML models (one based on RF to classify drives as good or in prefail state and one based on BiLSTM to predict the actual number of days left before failure).

In the process of creating this technical solution, through experimentation, a few suspicions were confirmed (one being that a ML model that should classify the state of a HDD needs to be trained on measurements collected from other drives of same model to achieve high precision), an observation of the impact the highly imbalanced nature of the dataset has on the results of the prediction models trained without the extra data preprossesing and balancing steps taken in other state-of-the-art works and (which were flagged as points of interest for future work) and a solution, that can indeed classify - with great accuracy and precision - the state of disks and predict their remaining life in days with a good confidence level, was delivered.

6.2 Future work

As expected, there are a few key issues that need to be addressed with the most important one being the highly imbalanced nature of the dataset which impacts the overall performance of the prediction models. Work needs to be done in order to perform balancing and normalization operations on the dataset in a practical way (preferably with the help of an automated process that does not require human intervention) to be able to get the precision of the RF model to values greater than 95% (while keeping its accuracy at least as high its precision) and to lower the MAE/MSE/RMSE metrics of the BiLSTM model such that they all are as close to 0 as possible and raise the value of R-squared to as close to 1 as possible (any value above 0.95 would be ideal).

Another important aspect that needs to be addressed in the future is the diversity of the data on which the models are trained. At this point in time the only publicly available dataset comes from Backblaze who use only a handful of disk models out of all options that are available to the public. This is easier said than done however with help from major service providers such as telecommunications companies and cloud service providers it can be achieved. The alternative would be to collect measurements directly from as many end users as possible (with the help of the pRUL agent). On a related note, a future version of the proposed technical solution will also be able to cover SSDs and NVMEs as more measurements are collected from various environments where these devices are in use.

Through experimentation an observation was made which is that the BiLSTM model is able to predict RUL when trained with longer lookback windows with better results however what remains unknow is what the limit is (what is the best lookback window?). At the same time work needs to be done such that RUL can be predicted more accurately using shorter lookback windows (by using new approaches to tackle the imbalanced nature of the dataset) such that users don't need to wait for a long time before they can get an accurate RUL prediction for their HDDs.

Chapter 7

Reflection

At its core, this work was born from the desire to tackle a problem that's been around for a while with storage in general which is that maintenance for disk drives has always been treated as a reactive operation (when a drive fails it needs to be replaced). Approaching it from a proactive point of view, where one observes a disk drive over time and based on the combination of certain SMART attribute values decides to replace a drive at a convenient time, ideally before it has the potential to cause an incident, has proven to be a good challenge. Many lessons were learned in the process of writing this thesis such as:

- how to identify good works in the field and what to extract from them
- how to isolate and define a specific problem that one wants to approach
- how to perform experiments, record findings, extract knowledge and identify next steps
- what it takes and how to write a report
- it takes a really long time and effort to train complex machine learning models
- you can't always reproduce the results and methods found in existing literature

The most difficult challenge faced while working on this thesis (other than the problem itself) was time, the pressure of meeting the deadline while at same time delivering on the promise. In this day of age, doing research for the sake of research is a luxury that few can afford and in this context the solution to the problem had to be shaped into the form of a smaller result that one can benefit from now rather than wait for a while longer and get a much better result later.

One particular challenge that was not overcome while working on this report is that of not being able to produce a method / algorithm that would aid in the data preprocessing stage of the work in an automated (scripted) / practical way with balancing the dataset. Another one, caused by the amount of time it takes to train models, is that more experimentation needs to be done with the BiLSTM model to find what the ideal lookback window should be as well as how many windows are needed to train a model that can produce excellent results. These will need to be addressed at a later time, ideally without pressure from deadlines.

The problem, at first, before it became a dissertation project, was formulated, in its simplified form, "let's try to predict disk failure by looking at measurements captured throughout their entire lifetime", then, after a bit of research into previous works the problem became "let's try to predict disk failure in a practical way" because right now it's not possible to learn from their entire lifetime. Both approaches started with the premise that in terms of accuracy / precision / confidence I could produce results comparable with state-of-the-art works in the field but, as I learned while working on the report, this is also not possible as these works focused on all sorts of intricate techniques to improve the quality of the data such that the models generate excellent predictions which are difficult to implement in real world production environments without hiring highly trained / specialised data scientists.

As a last note, this thesis together with all the work involved in addressing the problem at hand are publicly available on the Cristian Seceleanu (2023) website.

The short summary version of this chapter would be: "Challenge accepted!".

References

A. Bai, M. Chen, S. P. G. H. and Yang, Z. (2022), 'Attention-based bidirectional lstm with differential features for disk rul prediction [10]', *2022 IEEE 5th International Conference on Electronic Information and Communication Technology (ICEICT), Hefei, China* pp. 684–689.
URL: *https://ieeexplore.ieee.org/document/9908671*

A. Coursey, G. Nath, S. P. and Sengupta, S. (2021), 'Remaining useful life estimation of hard disk drives using bidirectional lstm networks [4]', *2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA* pp. 4832–4841.
URL: *https://ieeexplore.ieee.org/document/9671605*

A. De Santo, A. Galli, M. G. V. M. and Sperlì, G. (2022), 'Deep learning for hdd health assessment: An application based on lstm [2]', *IEEE Transactions on Computers, vol. 71, no. 1* pp. 69–80.
URL: *https://ieeexplore.ieee.org/document/9277641*

A multi-instance LSTM network for failure detection of hard disk drives [3] (2020), *2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, United Kingdom* pp. 709–712.
URL: *https://ieeexplore.ieee.org/document/9442240*

Backblaze (2016), 'What smart stats tell us about hard drives'. (accessed August 24, 2023).
URL: *https://www.backblaze.com/blog/what-smart-stats-indicate-hard-drive-failures/*

Backblaze (2023), 'Hard drive data and stats'. (accessed August 24, 2023).
URL: *https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data*

Cristian Seceleanu (2023), 'prul'. (all the work presented in this report is available publicly under a free license).
URL: *https://prul.seceleanu.co.uk*

F. D. d. S. Lima, G. M. R. Amaral, L. G. d. M. L. J. P. P. G. and d. C. Machado, J. (2017), 'Predicting failures in hard drives with lstm networks [7]', *2017 Brazilian Conference on Intelligent Systems (BRACIS), Uberlandia, Brazil* pp. 222–227.
URL: *https://ieeexplore.ieee.org/document/8247057*

F. D. S. Lima, F. L. F. Pereira, L. G. M. L. J. P. P. G. and Machado, J. C. (2018), 'Remaining useful life estimation of hard disk drives based on deep neural networks [6]', *2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil* pp. 1–7.
URL: *https://ieeexplore.ieee.org/document/8489120*

Lihan Hu, Lixin Han, Z. X. T. J. H. Q. (2020), 'A disk failure prediction method based on lstm network due to its individual specificity [5]', *Procedia Computer Science Volume 176* pp. 791–799.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050920319700*

QNAP (2023), 'Da drive analyzer'. (accessed August 21, 2023).
URL: *https://www.qnap.com/en/software/da-drive-analyzer*

S. Basak, S. S. and Dubey, A. (2019), 'Mechanisms for integrated feature normalization and remaining useful life estimation using lstms applied to hard-disks [8]', *2019 IEEE International Conference on Smart Computing (SMARTCOMP), Washington, DC, USA* pp. 208–216.
URL: *https://ieeexplore.ieee.org/document/8784059*