Deep Learning for HDD Health Assessment: An Application Based on LSTM

Aniello De Santo, Antonio Galli, Michela Gravina, Vincenzo Moscato, and Giancarlo Sperlì

Abstract—Hard disk drive failures are one of the most common causes of service downtime in data centers. Predictive maintenance techniques have been adopted to extend the Remaining Useful Life (RUL) of these drives, and minimize service shortage and data loss. Several approaches based on machine and deep learning techniques have been proposed to address these issues, mostly exploiting models based on Self-Monitoring analysis and Reporting Technology (SMART) attributes. While these models have proven to be reliable, their performance is affected by the lack of information about the proximity of disk failure in time. Moreover, many of these techniques are sensitive to the highly unbalanced nature of existing data-sets, in terms of good to failed hard disk ratio. In this article, we propose a LSTM (Long Short Term Memory)-based model combining SMART attributes and temporal analysis for estimating a hard drive health status according to its time to failure. Our approach outperforms state-of-the-art methods when evaluated on two data-sets, one containing hourly samples from 23 395 disks and the other reporting daily samples from 29 878 disks. Experimental results showed that our approach is well suited to data-sets with different sampling periods, being able to predict hard drive health status up to 45 days before failure.

Index Terms—Hard drive failure prediction, SMART, health degree, long short-term memory

1 Introduction

Hard disk drives (HDD) are nowadays a primary type of storage in data centers. Due to this pervasive use, HDD failure is now one of the main factor for data center downtime, unavailability, and data loss — with obvious effects on overall business costs and reliability. Thus, an important line of research has focused on developing robust predictive maintenance techniques, to reliably predict HDD failures, and timely adopt maintenance strategies to increase the *Remaining Useful Life* (RUL) of these drives.

HDD *maintenance* actions include inspection, testing, repair, and replacement — basically any action aimed at preserving the quality of the system while improving its availability and extending its life. Obviously, in order for maintenance costs for large distributed systems to remain effective, such actions have to be properly organized according to well-defined strategies. Because of the dimensions of most data-centers, properly scheduling these actions is not a trivial problem, and maintenance policies have become a fruitful topic of research [1], [2, a.o.].

Recently, the dissemination of condition monitoring equipment and the development of methods for deterioration prognosis and residual life estimation have shifted the interest of most practitioners towards *predictive maintenance* techniques [3]. Predictive maintenance seeks to anticipate system failures in order to plan timely interventions on the system. In such frameworks, decisions are based on a system's online health prognostic information (i.e., information about the system future state), rather than on the online diagnostic information (i.e., information on the system current state) [4].

Due to this shift in focus towards predictive systems, machine learning approaches have been gaining increasing popularity [5, a.o.]. In particular, models based on *Self-Monitoring, Analysis and Reporting Technology* (SMART) have shown high accuracy levels by relying on internal attributes of HDDs as indicators of drive reliability.

Importantly, most prediction systems analyze HDD failure as a binary classification task, simply distinguishing between good hard drives and those at high risk of failure. However, the complexity of the prediction task and the unbalanced nature of the data used in training have shown how these models' performance goes down significantly when they are tested on data-sets representative of real-world environments [6].

Moreover, it has been suggested that, due to the nonlinear pattern dynamics found in real-world systems, the ability to model the proximity of possible failures in time (and not just the chance of failure) could fundamentally change the way maintenance strategies are optimized [7]. Following this intuition, in this paper we implement a HDD health level prediction task that models the health degree of a HDD unit according to its estimated time to failure. In particular, we propose a framework that leverages a *Long Short Term Memory* model for HDDs' health level prediction. Its main characteristics can be summarized as follows:

- it automatically identifies the HDD health levels by considering the distribution of SMART attribute values over the time;

- *Aniello De Santo is with the Department of Linguistics of the University of Utah, Salt Lake City, UT 84112 USA. E-mail: aniello.desanto@utah.edu.*
- *Antonio Galli, Michela Gravina, Vincenzo Moscato, and Giancarlo Sperlí are with the Department of Electrical and Information Technology of University of Naples Federico IIvia Claudio 2180125 Naples, Italy. E-mail: antonio.galli@unibg.it, {michela.gravina, vincenzo.moscato, giancarlo.sperli}@unina.it.*

Manuscript received 5 Nov. 2019; revised 10 Nov. 2020; accepted 29 Nov. 2020. Date of publication 2 Dec. 2020; date of current version 13 Dec. 2021.
(Corresponding author: G. Sperlí.)
Recommended for acceptance by L. Zhou.
Digital Object Identifier no. 10.1109/TC.2020.3042053

- it improves prediction accuracy by considering sequential dependencies in SMART attributes;
- it relies on an automated strategy for identifying the number and size of hard drive's health degree settings.

First, we implement an automated step for HDD health level definition using a Regression Tree (RT) algorithm. We then exploit LSTM networks [8] to model the sequential dependencies between SMART attributes over time. LSTMs are particularly appropriate for this task, as they were explicitly designed to model long-range dependencies in temporal sequences.

While LSTM approaches to RUL estimation have been successfully explored in the past [7], [9], [10], our methodology proves more flexible to the highly complex nature of the data by relaxing the predefined health degree levels traditionally used in the literature in favor of dynamically generated ones. Identifying HDD health levels automatically allows us to take full advantage of the information available in the training sets, and to obtain finer-grained predictions beyond what would be available through the simple binary classification task used in current systems.

In order to support the efficacy and practicality of our model in real-world scenarios, we evaluate its performance over two data-sets. For each hard drive in our data-sets, we extract attribute sequences from specific time windows (TW) of varying size (from 4 to 48 hours for the fist data-set and from 5 to 14 days for the second ones), and show how this approach outperforms a variety of models and methods in the previous literature [4], [11], [12, a.o.].

The rest of the paper is organized as follows: Section 2 motivates the research enterprise and clarifies current issues concerning HDD maintenance; Section 3 is a summary of related work; Section 4 introduces the proposed approach; Section 6 describes the data-set and the experimental setup used for evaluation; Section 7 discusses our results and model comparisons. Section 8 concludes with an outline of possible future work.

2 HDDs and Predictive Maintenance

Introduced by IBM in 1956, HDDs have become a reliable, wide-spread technology for data storage. So wide-spread, in fact, that in 2017 Western Digital predicted that by 2020 70% of all data would be stored in HDDs [13]. As of March 2019 Backblaze — a pioneer data storage provider — reported 106,238 spinning hard drives in their cloud storage ecosystem spread across three data centers [14]. Even the rise of modern Solid State Drives (SDDs) — a storage technology with no moving parts which instead uses semiconductor chips with storage cells — has not affected the popularity of HDDs in data-centers. There is a simple reason for this: although SSDs consume less energy, HDDs' trade-off between storage capacity, life expectancy, and cost remains unbeaten [15].

While the pervasive use of HDD technology is thus unquestionable, its reliability remains an important issue. Moreover, the increasing popularity of big data applications has made it so that storage systems are required to possess exabytes of capacity, usually resulting in millions of hard disk drives per data center [16]. Obviously, at such a scale disk failures become the norm. The fundamental concern is then not just the life-expectancy of a standard HDD, nor the quality of information storage, but the frequency of unexpected failures. Therefore, resources have recently been focused on the search for optimal strategies to deal with such failures.

In order to come up with cost-effective strategies against data-loss, it is important to individuate the primary causes of disk failures. Being aware of these causes is the first step towards mitigating HDD failure risk, as it allows for a variety of monitoring systems that target each failure source specifically.

As disk failure in large-scale storage systems becomes unavoidable though, solutions to data-loss have increasingly been relying on redundant arrays of inexpensive disks (RAID; [17]). However, RAID recovery is a time-consuming process which requires bringing the failed system offline for repair. This results in delays on the user side — as the number of data sources decreases considerably while the damaged disk is being replaced — and in additional stress on the remaining disk drives due to the intensive read and write activities [18]. Therefore, while RAID approaches are an effective way to address the data-loss problem from a general perspective, they remain an overall unsatisfying solution.

Importantly, these kind of methods are *reactive*: they provide a safety net in case a failure occurs. Reactive storage protection approaches thus suffer from high recovery overheads, which significantly affect data availability and system performance [18]. For such reasons, recent efforts have been focused on exploring *proactive* solutions to HDD failure. If we have technologies that are able to predict disk failure in advance with a high confidence rate, then we can plan data rescue operations so that they overlap with regular storage operations — thus reducing the intervals of data-unavailability. Effectively predicting HDDs' health status becomes then the core challenge of maintenance operations.

In this sense, predictive maintenance models have become the state-of-the-art in maintenance research, by using historical logs of real-time data to predict the future failure of still operating hard disks.

In the last few years, predictive technologies have been relying more and more on machine learning techniques grounded in big data analysis in order to monitor the online health status of a hard drive, and improve fault prediction accuracy. Importantly though, this comes with the challenge of balancing information coming from multiple data sources (e.g., temperature, vibration), in order to take into consideration all of the possible causes of failure, and come up with appropriate maintenance strategies. Furthermore, it has been observed that hard drives often deteriorate gradually over time [2, a.o.]. Thus, it is important to be able to model the temporal dynamic of the dependencies within SMART attributes.

However, predictive systems need to strike a careful balance when extrapolating health information from these sources in order to raise failure warnings, as high false alarm rates would lead to as many overhead costs as missed disk failures. It is then crucial to find the most sensible way to exploit each HDD's health attributes, so to balance high

TABLE 1
Overview of State-of-the-Art approaches

	Methodology	Pros	Cons
[5]	A multiple-instance learning framework using a naive Bayesian classifier for predicting failures	Maintaining low false alarm rate.	No information about HDD health level status. The main focus is on false alarm avoidance.
[11]	A classification and regression model based on SMART attributes for predicting failures	Maintaining low false alarm rate.	No information about HDD health level status. The main focus is on false alarm avoidance.
[4]	An RNN model based on SMART attributes for evaluating HDD status.	Performs health status assessment.	Health degree settings manually defined in terms of number and size of each interval. The prediction phase combines the prediction obtained by the analysis of the last 1 hour.
[3]	A prediction model based on a part-voting Random Forest algorithm.	The prediction model differentiates failure prediction in a coarse-grained manner	The model does not consider data correlation and historical information about HDDs.
[20]	A prediction model based on an Online Random Forest algorithm.	The model evolves with sequential arrival of data on the flying adapting to SMART distribution over the time	This model does not consider data correlation and historical information about HDDs.
[21]	A two-step approach: anomaly detection according to a sliding window and a failure prediction model.	It tries to balance the failure detection rate and false alarm.	It does not provide any information about HDD health assessment. The sliding window is manually computed based on the number of samples ignoring correlation with the timestamp.
[22]	Compares two model based respectively on Random Forest and LSTM for HDD RUL estimation.	It predicts an HDD's Remaining Useful Life.	Health degree settings manually defined in terms of number and size of each interval. Only the current snapshot of SMART attribute values's sequence are considered according to the number of samples ignoring correlation with the timestamp.
[7]	An LSTM-based prediction model for Remaining Useful Life extimation.	It performs a hyper parameter optimization with to improve the prediction phase.	Health degree settings manually defined in terms of number and size of each interval. Only the current snapshot of SMART attribute values' sequence are considered according to the number of samples ignoring correlation with the timestamp.
[9]	A method based on adversarial training and layerwise perturbation for HDD health status prediction.	They propose a Layerwise Perturbation-Based Adversarial Training method to deal with overfitting and biased fitting problems.	Health degree settings manually defined in terms of number and size of each interval.
[23]	Pipeline based on SMART attributes for disk replacement.	Statistical technique are used to correlate SMART parameters and disk replacement.	The authors consider only two classes (healthy and replaced) and they do not provide any information about the Remaining Useful Life.
[24]	A cost-sensitive ranking-based machine learning model for disk error prediction.	Combining SMART attributes with system-level signals.	The authors consider only two classes and they do not provide any information about the Remaining Useful Life.
[25]	A TCNN-based prediction model for hardware failure prediction.	TCNN model for analyzing SMART attribute distribution over time.	The authors consider only two classes and they do not provide any information about the Remaining Useful Life.

accuracy and conservative predictions. This requires expert domain knowledge, in order to best tune the features of the model and address challenged due to feature specificity and cross-correlation between health attributes.

Moreover, hard disk data vary between manufacturers — and even between hard disk models within the same manufacturer. This implies that a previously specified model cannot be carried over to perform equally well in different data centers, and explicit feature engineering needs to be repeated multiple times based on the characteristics of each model [19].

In this paper, we follow recent research in predictive maintenance, and present a deep learning approach addressing many of the current problems in the literature (data sparsity, need for domain knowledge, manual feature engineering).

3 Related Work

As mentioned, HDD failure prediction plays a very important and crucial role in reducing data center downtime and significantly improving service reliability. By collecting information about the *health* conditions of HDD in real-time, SMART records have been consistently used in failure detection systems — though with remarkably low failure detection rates (FDRs).

The use of information collected by sensors online is clearly essential to efficient prediction systems. Thus, different methods have been proposed to optimize the performance of SMART based models: from Bayesian classifiers [1] and support vector machines (SVM; [5]), to classification trees [11], back-propagation neural networks (BPNNs; [12]), and recurrent neural networks (RNNs; [4]). A summary of the approaches analyzed in this Section is presented in Table 1.

Hamerly & Elkan [1] were among the first to use Bayesian modelling in failure prediction. The intuition behind their work is to re-frame failure prediction as an anomaly detection problem, and then classify test data based on the probability of reading from an HDD behaving normally, and information about the HDD internal conditions. They compare two methods — a mixture of Naive Bayes submodels trained on expectation-maximization (EM) and a Naive Bayes — and show how their models perform significantly better than previously industry level predictors.

Following this line of investigation, Murray *et al.* [5] conduct an extensive evaluation of four different methods (SVM, unsupervised clustering, rank-sum and reverse arrangements test). They then propose an algorithm based on the multiple-instance learning framework and a naive Bayesian classifier (*mi-NB*) to deal with false-alarm rate

(FAR). Also interested in minimizing FAR while maximizing detection accuracy, Zhu *et al.* [12] evaluate the accuracy of an SVM and a back-propagation neural network exploiting SMART attributes *and* their change rates. They show that while the SVM model achieves the lowest FAR (0.03%), the back-propagation model has the best high detection rate (95%). A similar approach is adopted by Li *et al.* [11], which propose a surprisingly well-performing prediction model based on regression trees.

More recently, Xu *et al.* [4] suggested that the inefficiency of past prediction systems stems from the fact that most HDD are not simply *good* or *bad*, but they are subject to gradual decay. Thus, they use Recurrent Neural Networks (RNNs) to model gradual changes in sequential SMART attributes and better capture the shifting nature in HDDs' health status. Their model achieves better performance than previous sequence independent models *and* short-term sequence dependent models. With a similar goal in mind, Botezatu *et al.* [23] designed a pipeline based on SMART attributes for predicting disk replacements approximately 15 days in advance. To deal with over-fitting and biased-fitting problems, [9] propose a Layerwise Perturbation method, using adversarial training to predict HDD health status on the basis of three health levels manually defined. Furthermore, a cost-sensitive ranking-based machine learning model, combining SMART attributes with system-level signals, has been proposed by Xu *et al.* [24] for predicting disk error.

Finally, it has been observed that the data available on HDDs' health status is highly unbalanced (in particular, in the ratio of healthy and failure samples) — an imbalance that is bound to affect the performance of prediction system relying on intrinsic features.

To partially address this issue, Shen *et al.* [3] propose a failure prediction model based on a part-voting random forest algorithm which compensates for data unbalance using a clustering-based under-sampling method.

Similarly, Xiao *et al.* [20] exploit an online random forest prediction model, which evolves on-the-fly with sequential arrival of data, according to the variance of SMART distribution over time. Their model addresses both the problem of labelling sequential samples gathered on-the-fly, and the high unbalance in the distribution of healthy/failing disks.

Sun *et al.* [25] propose the use of a temporal Convolutional Neural Network (TCNN) in order to address the high variability of delay-to-failure values in real world scenarios with sparse failure samples, while reducing sensitivity to noise in the analysis of SMART distributions over time. To address issues due to data sparsity, they extend the binary cross-entropy loss function emphasizing the loss of misclassified samples. TCNNs within this approach show superior performance than RNNs and LSTMs.

Similarly efficient methods of failure prediction have highlighted the fundamental importance of taking into account the sequential nature of the SMART attributes when modelling failure rates [21]. In this spirit then, it seems to be crucial to define models that adapt to the dynamical changes in the SMART attributes, while taking into account the unbalanced distribution of the training data [2]. In this sense, LSTMs seem to be especially up to the task, as they have been shown to be sensitive to long-term dependency and the dynamical nature of time-series data across a variety of domains [26], [27], [28].

Thus, we follow recent work on predictive models for HDD failure using neural-networks [7], [22], and propose a LSTM based model for HDDs' health level prediction task, which automatically identifies the HDD health levels.

We evaluate our model on two SMART data-set (23,395 and 28,878 HDDs), and perform a comparative evaluation with a set of other models in the literature.

4 Methodology

Since hard drives often deteriorate gradually rather than abruptly, we argue that temporal analysis methods should be employed to model the sequential nature of the dependencies within SMART attributes over time. Thus, we propose an approach to estimate the Remain Useful Life (RUL) of a HDD, by automatically identifying specific health conditions on the basis of SMART attributes values. This methodology is grounded in three main steps:

- *Hard Drive Health Degree Definition*. In which a status (or health level) is defined for each hard drive according to its time to failure;
- *Sequences Extraction*. In which sequences in a specific time window are extracted for each hard drive;
- *Health Status Assessment Through LSTM*. In which a health level is associated to each temporal sequence.

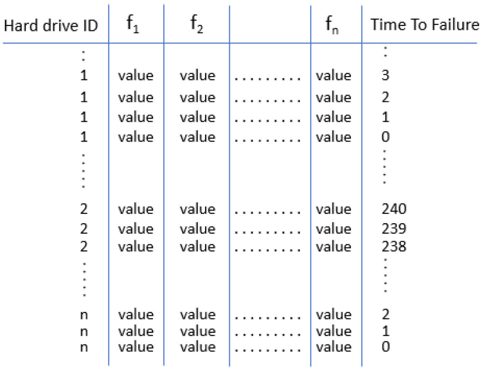
In what follows, we describe each component of our framework in detail.

4.1 Health Degree Definition

Hard drive failure in real-world data centers is a gradual process of deterioration. To address the gradient nature of the decay, we follow [4] in defining the health status (or level) of a HDD according to its time before failure. Differently than [4], we implement an automated step for HDD health level definition.

More specifically, in this step we consider only the hard drives that are going to fail, introducing for each of them an additional feature representing the *time before failure*. The data-set reports, for each hard disk, the temporally sorted sequence of SMART attributes with a specific sampling period. Denoting with m_j be the number of samples for the hard disk j, it is possible associate each sample with an index i from 0 to $m_j - 1$, representing the number of samples that follow it in the sequence describing hard disk failure. As a consequence, the sample with index $i = 0$ is the last sample before failure. In Fig. 1, *Time-to-failure* is the feature representing the time before failure for each hard drive whose meaning depends on sampling period while f_1, f_2, ..., f_n are the SMART attributes.

Our idea is to build a Regression Tree (RT) for each SMART attribute f_i with $i = 1, 2 \dots n$, having the feature representing the time before failure as predictor and f_i as the numeric target value. Among all the resulting trees (one for each SMART attribute f_i), the one with the highest performance is selected, showing the attribute most temporally dependent. Since the selected Regression Tree (RT) presents splits only on the feature *Time-to-failure*, the latter is used to distinguish hard drive health levels according to time before



Hard drive ID	f_1	f_2		f_n	Time To Failure
⋮					⋮
1	value	value	....	value	3
1	value	value	....	value	2
1	value	value	....	value	1
1	value	value	....	value	0
⋮					⋮
2	value	value	....	value	240
2	value	value	....	value	239
2	value	value	....	value	238
⋮					⋮
n	value	value	....	value	2
n	value	value	....	value	1
n	value	value	....	value	0

Fig. 1. *Time to failure* is a feature representing the time before failure for each hard drive sample, while f_1, f_2, ..., f_n are the SMART attributes.

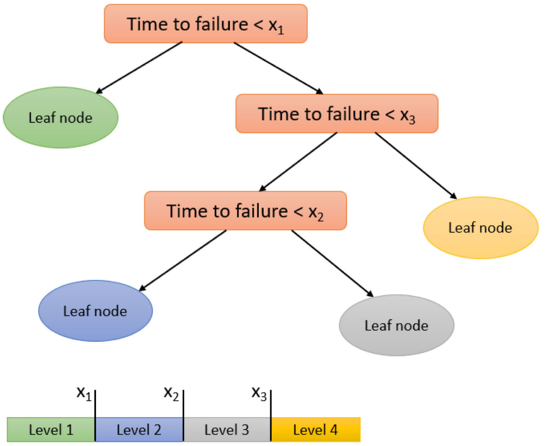


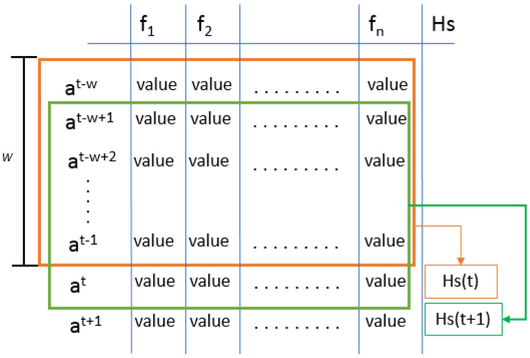
Fig. 2. Identification of hard drive health levels by means of a Regression Tree algorithm. Each internal node represents a split on the feature Time-to-failure, resulting in the definition of four health degree levels.

failure. Fig. 2 is an example of hard drive health levels identification by means of the Regression Tree algorithm. Each internal node represents a split on the feature Time-to-failure, resulting in the definition of four health degree levels.

As mentioned above, the automated step for health-level definition only considers those hard drives that are going to fail. A different level or status should be assigned to samples belonging to hard drives that will not fail since they have been excluded in this step. More specifically, the samples belonging to the hard drives that will not fail are labelled as *Good*.

4.2 Sequence Extraction

To explore the temporal dependencies within the SMART features periodically collected for each hard drive, we extract feature sequences in specific time windows (TW).



	f_1	f_2		f_n	Hs
a^{t-w}	value	value	....	value	
a^{t-w+1}	value	value	....	value	
a^{t-w+2}	value	value	....	value	
⋮					
a^{t-1}	value	value	....	value	
a^{t}	value	value	....	value	Hs(t)
a^{t+1}	value	value	....	value	Hs(t+1)

Fig. 3. Sequence extraction step for a single hard drive.

Let w and a^t be the time window size and the set of SMART features $(f_1, f_2 \dots f_n)$ at time t, respectively. Our model aims to predict hard drive health status at time $t+1$ $(Hs(t+1))$ considering the sequence $(a^{t-w+1}$..., a^{t-1}, $a^t)$. For each a^t, the health status $Hs(t)$ is defined as the Regression Tree built according to in Fig. 4, and the feature sequence for each hard drive at time t is extracted considering the $w-1$ previous samples (cf. Fig. 3). Each sequence results in a bidimensional array of size $w \times n$, where n is the number of SMART features considered. For each hard drive, sequences are extracted with a stride of one. It follows that $m_j - w + 1$ sequences are extracted for each hard drive, where m_j is the number of samples for the disk j.

For each sequence $(a^{t-w+1}$..., a^{t-1}, $a^t)$, the hard drive's health level is defined by the health level of the set of features a^{t+1}. The result of this step is a sequence-based dataset — a set of bidimensional arrays, each associated to a health level representing the hard drive's health condition between two consecutive samples (i.e., a^t and a^{t+1}).

4.3 Health Status Assessment Through LSTMs

Based on what we established in the previous sections, it should be clear how hard drives' health level prediction consists in a multiclass classification task, where each feature sequence is assigned to one of the classes (health levels) introduced in Section 4.1.

Because of the sequential, gradiently changing nature of the SMART features, it is important that our model is able to capture dependencies across features over time. Long Short Term Memory networks (LSTMs) are extension to recurrent neural networks, explicitly designed with the purpose of learning long-term dependencies [8]. They are widely used nowadays, as they work tremendously well on a large variety of problems.

In our framework, the input to each LSTM layer is a three-dimensional data structure of size $z \times w \times n$, where:

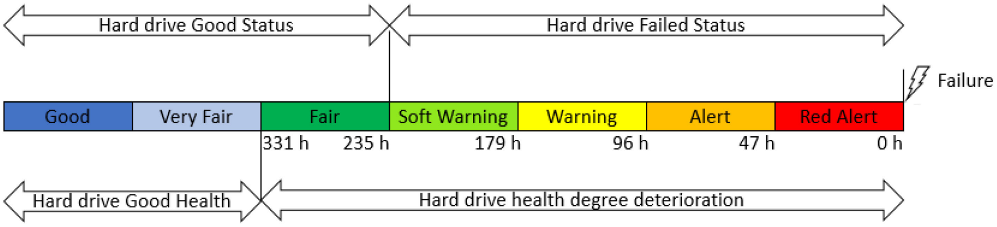


Fig. 4. Hard drive's health degree settings for the Baidu data-set.

- z is the the total number of sequences (or the batch size at each iteration);
- w is the size of each sequence — that size of a time window, in terms of time steps;
- n is the total number of features describing each time step.

Since the percentage of failed hard drives is often small compared to the percentage of good hard drives, the sequence extraction step may result in an unbalanced data-set with the majority of sequences belonging to the *Good* level. As a consequence, we introduce a data balancing step, so that the input to the network is a set of balanced data.

In particular, the sequence-based data-set is balanced by replicating the sequences belonging to the minority classes. We argue that sequences replication is an efficient balancing strategy that avoids the polarization of the classification model on a single class without creating synthetic data or reducing the data-set size by sampling the instances beloging to the majority class. The implemented classification network has two stacked LSTM layers with 128 units, followed by a single dense layer.

5 Architecture

We deploy our approach according to the *lambda* architecture pattern [29]. We distinguish three main phases: *ingestion*, *data management*, and *processing*.

In the first stage, we use the stream processor *Apache Kafka*[1] to collect SMART attributes about each disk. We assign a timestamp to each attribute sequence, that is successively stored into the NoSQL database Apache Cassandra.[2] We chose Cassandra due to the low running-time associated to its write operation. In particular, each stored tuple is composed by HDD's identifier, timestamp and a sequence of SMART attributes values.

In the third phase, the stored data is processes according to our time window approach, with the deep learning component implemented using Keras.[3] Specifically, the classification network is implemented as two stacked LSTM layers with 128 units, followed by a single dense layer with number of units equal to classes number, and softmax as a activation function.

6 Experimental Evaluation

This section aims to evaluate the effectiveness of our approach. We test the prediction performance of the model on the two different SMART data-set, and then compare our performances with those of three other methods explored in the literature: a Classification Tree model, a Random Forest model, and a model based on Multiclass Neural Networks.[4]

6.1 Baidu Data-Set

The first SMART data-set used for our analysis was collected from a single running data center of Baidu Inc,[5] and contains samples from 23,395 disks. All samples refer to an enterprise-class disk model of Seagate (ST31000524NS). Each disk was labeled *Good* or *Failed*, with only 433 disks in the failed class and the rest of disks (22,962) in the good class. As the ratio of *Good* versus *Failed* disks is approximately $1:50$, we consider this a highly unbalanced data-set.

TABLE 2
SMART Attributes as Features

SMART ID #	Attribute Name
	Serial Number
	Label
1	Raw Read Error Rate
3	Spin Up Time
5	Reallocated Sectors Count
7	Seek Error Rate
7	Power On Hours
187	Reported Uncorrectable Errors
189	High Fly Writes
194	Temperature Celsius
195	Hardware ECC Recovered
197	Current Pending, Sector Count
5	Raw Value of Reallocated Sectors Count
197	Raw Value of Current Pending Sector Count

SMART attribute values were read per-hour for each disk. For *Good* disks, every sample collected over a week is kept in the data-set, so every good disk is associated to 168 samples. For *Failed* disks, samples in a longer time period (20 days before actual failure) are saved, resulting in a maximum of 480 samples per disk. Note though that a specific disk could actually be associated to a smaller number of samples, if it failed during the 20 days of operation since the start of data collection. Finally, each entry in the data-set contains the 14 features listed in Table 2, with values for every attribute value scaled to the same interval $[-1, 1]$.

6.2 Backblaze Data-Set

The Backblaze data-set[6] contains daily data collected from 50,984 hard disks. Each sample consists of information about timestamp, disk serial number, disk model, disk capacity and values for 90 SMART attributes. Moreover, for each sample the feature *failure* is set to 0 if the drive is alive while it is set to 1 if the disk has been replaced the following day. We excluded all samples before February 2014, since more than 70% of SMART parameters had not been collected. We focused on samples belonging to Seagate ST4000DM000, since it is the most populated model in data-set (29,878 disks in total; 29,083 *good* disks and 795 *failed* disks). Among all the SMART attributes, we selected the ones that are shared between the Backblaze and the Baidu data-sets (see Table 2). However, we also excluded the SMART attribute with ID 195 (*Hardware ECC Recovered*), since no sample had a value associated to this feature. Finally, the values for every SMART attribute were scaled to the interval $[-1, 1]$.

[1] [Online]. Available: https://kafka.apache.org/

[2] [Online]. Available: http://cassandra.apache.org/

[3] [Online]. Available: https://keras.io/

[4] [Online]. Available: To encourage reproducibility of our results, all code is publicly available on Github (https://github.com/Anthonys94/HDD_deep_learning).

[5] [Online]. Available: http://pan.baidu.com/share/link?shareid=189977&uk=4278294944

[6] [Online]. Available: https://www.backblaze.com/b2/hard-drive-test-data.html

6.3 Preprocessing

Data preprocessing consisted of two main steps:

6.3.1 Features Selection

For both data-sets, the features the We also excluded the feature representing disk capacity in the Backblaze data-set. Importantly, the attributes *Label* for Baidu, *failure* for Backblaze, and *Serial Number* for both data-sets are necessary in order to distinguish between failed and good hard drives and to create sequences for each hard drive. However, they are not taken into account during sequence classification.

For good hard drives, each sample was associated to the health degree level *Good*, while for failed hard drives, their remaining functioning time depends on the number of samples collected for said device.

6.3.2 Health Degree Computation

The way health degree levels are computed differs between the Baidu and the Backblaze data-sets.

- *Baidu Data-Set*. For *failed* disks, stored samples correspond to a period of 20 days before actual failure. Thus, we propose a model for predicting hard drive health status 20 days in advance. As mentioned in Section 4.1, hard drive health degree definition depends on the splits of the selected Regression Tree (RT) model on the feature Time-to-failure. Recall that the Baidu data-set presents samples read per-hour for each disk. For this reason, we renamed the feature Time-to-failure *Hour to failure*. We selected the regression tree built with the feature *Raw value of Current Pending Sector Count* reported in Fig. 5. Specifically, the selected Regression Tree suggests distinguishing 6 different levels of *health degree* for hard drives that will fail. We then introduced a different level for those hard drives that will not fail. This results in the definition of 7 levels, named as follows:
  - *Good*. The hard drive works properly. This level is associated to samples belonging to hard drive that will not fail;
  - *Very Fair*. The hard drive works properly, but a fault or an error may have occurred;
  - *Fair*. The health status of the disk drive is fair and the hard drive is probably going to fail in less that 332 hours (approximately 14 days);
  - *Soft Warning*. The hard drive is going to fail in less that 235 hours (approximately 10 days);
  - *Warning*. the hard drive is going to fail in less that 179 hours (approximately 7 days);
  - *Alert*. the hard drive will fail in less than 96 hours (approximately 4 days);
  - *Red Alert*. the hard drive will fail in less than 47 hours (approximately 2 days).

  The levels *Good* and *Very Fair* represent HDDs still in good health conditions. They both imply that a hard drive works properly, and thus time constraints are left unspecified. The other statuses are associated with different degrees of deterioration. Therefore, we classify a hard drive as being in a *Good status*, if its health level is characterized as *Good*, *Very Fair*, or *Fair*. A hard drive is classified as being in a *Failed Status*, if its health level is in *Soft Warning*, *Warning*, *Alert*, or *Red Alert*. Fig. 4 shows the health degree settings for a single hard drive, as used in our evaluation. By assumption, the health degree level *Good* in never assigned to failed hard drives because of the high probability of errors or faults. Thus, the distinction between good and failed hard drives is preserved.

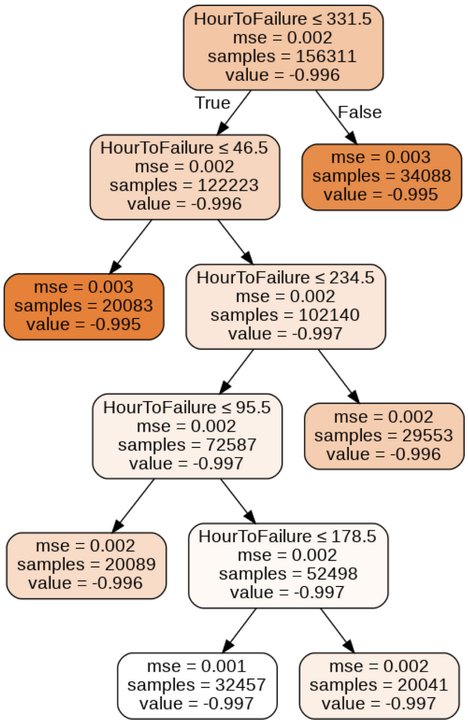
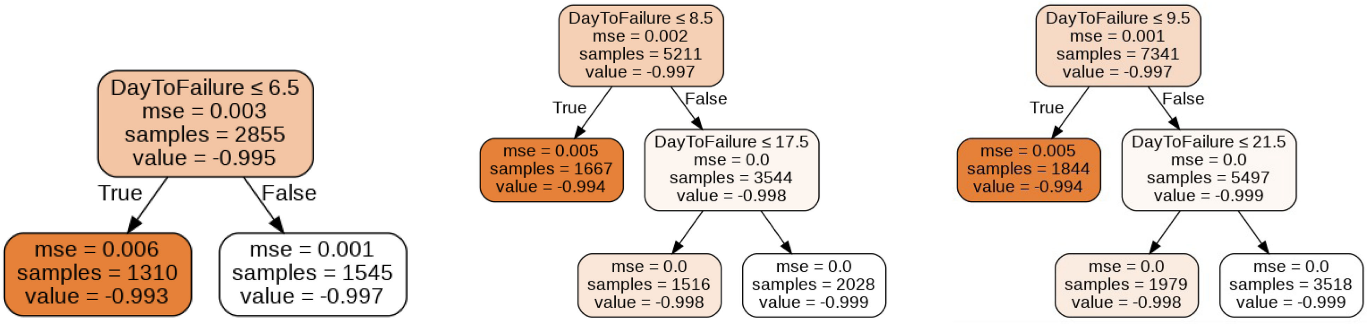


Fig. 5. Identification of hard drive health levels by means of the Regression Tree algorithm built on the feature *RawCurrentPendingSectorCount* for the Baidu data-set. For each leaf node *mse* is the mean squared error of the samples, *samples* is the number of samples in that node, and *value* is the value of the SMART attribute f_i for the samples in that leaf. For each internal node, we report the condition on the feature *Hour to failure*.

- *Backblaze Data-Set*. Since samples were collected from February 2014 to December 2015, failed disks present a long observation period. As a consequence, there is a high probability that at the beginning of that period, the disks having samples for more than one year were *good disks*. We argue that it is not possible to determine the exact time of error occurrence. For this reason, we focused on the last q samples of each failed hard drives, where q is a *prediction window* that determines the period in which hard drive health status should be assessed. Specifically, our approach is able to predict hard drive health status q days before failure.

  We explored different values for q, from 15 to 45 days. After choosing the value for q, hard drive health levels are defined according to Section 4.1. Since the Backblaze data-set contains daily samples for each hard drive, the feature *Time-to-failure* has been renamed *Day to failure*. We then selected the regression tree built with the feature *Raw value of Current Pending Sector Count*.

  Fig. 6 shows the regression trees obtained by selecting $q = 15$, $q = 30$ and $q = 45$. More specifically,



(a) Regression tree obtained with *q=15* **(b)** Regression tree obtained with *q=30* **(c)** Regression tree obtained with *q=45*

Fig. 6. Identification of hard drive health levels by means of the Regression Tree algorithm built on the feature *RawCurrentPendingSectorCount* for the Backblaze data-set. For each leaf node *mse* is the mean squared error of the samples, *samples* is the number of samples in that node, and *value* is the value of the SMART attribute f_i for the samples in that leaf. For each internal node, we report the condition on the feature *Day to failure*.

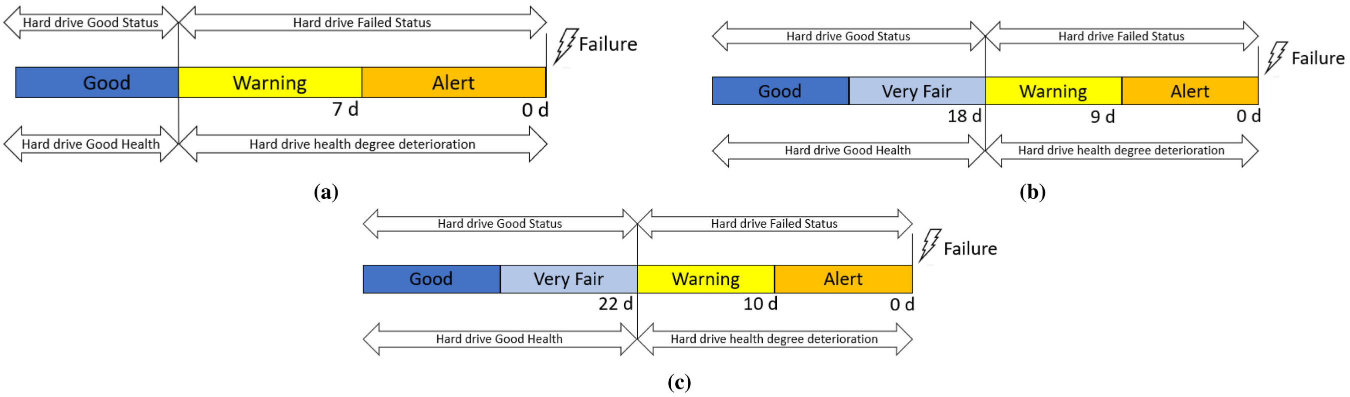


Fig. 7. Hard drive's health degree settings for Backblaze data-set for *q=15* (a), *q=30* (b), and *q=45* (c).

Figs. 6b and 6c suggest distinguishing 3 different levels of *health degree* for hard drives that will fail, while Fig. 6a suggests 2 levels. We then introduced a different level for those hard drives that will not fail. When q is set to 30 or 45, the result is the definition of 4 levels, labelled *Alert*, *Warning*, *Very Fair* and *Good*. In turn, if q is set to 15, we define 3 levels, labelled *Alert*, *Warning* and *Good*. The levels *Good* and *Very Fair* represent HDDs still in good health conditions. Therefore, we classify a hard drive as being in a *Good status*, if its health level is characterized as *Good* or *Very Fair* while a hard drive is classified as being in a *Failed Status*, if its health level is in *Warning* or *Alert*. Fig. 7 shows the health degree settings for a single hard drive, as used in our evaluation. Similary to the Baidu data-set, the health degree level *Good* is never assigned to failed hard drives due to the high probability of errors or faults.

6.4 Experimental Setup

As discussed above, we propose an automatic step for hard drive health levels definition, building a Regression Tree (RT) for each SMART attribute f_i, with the feature representing the time before failure as predictor. The selected trees (see Figs. 5 and 6) consider the SMART attribute *Raw Value of Current Pending Sector Count* as numerical target value. The function measuring the quality of a split is the mean squared error (*mse*). The minimum number of samples required for leaf node in the Regression Tree is 20000 for the Baidu data-set and 1830, 1380, and 1200 for the Backblaze data-set with $q = 45$, $q = 30$ and $q = 15$ respectively.

We evaluate our approach with respect to three of the *sequence independent* methods most used in the literature: a Classification Tree (CT), a Random Forest (RF), and a Multiclass Neural Network (MNN) — a deep neural network with dense layers. These models are sequence independent because they generalize over input samples rather than sequences, and thus don't take the temporal dependencies of the SMART attributes into account. Downstream of the parameters optimization, the number of trees for RF is set to 110 and 210 for the Baidu data-set and the Backblaze data-set respectively, and the minimum number of samples required for leaf node in CT is 20 for both data-sets.

We implement the RT, CT and RF models using the Python scikit-learn package, and we use Keras with Tensorflow as the backend for LSTM and Multiclass NN models.

As standard for this kind of techniques, the original SMART data-set was divided into training, validation and test sets. More specifically, we take the 70% of the data as training set, the 15% as validation set and the remaining data as test set.

During the training phase of the LSTM and Multiclass NN models, the maximum number of epochs is set to 150, and the batch size to 500. As an optimizer, we use Adam [30] with learning rate set to 0.001.

6.5 Performance Evaluation

Since each sequence is associated with one of the levels presented in Section 4.1, the HDD's health level assessment as

TABLE 3
Performance Values for the LSTM Models Obtained by Varying TW Size on the Baidu Data-Set

TW SIZE [hour]	*Accuracy*	*Precision*	*Recall*	ACC_G	ACC_F	ACC_G^{TOL}	ACC_F^{TOL}	*FDR*	*FAR*
48	**99.80%**	**99.1%**	**98.9%**	**99.83%**	**93.17%**	**99.89%**	**98.31%**	**98.2%**	**0.2%**
36	98.78%	98.8%	98.7%	99.80%	91.89%	99.87%	97.45%	97.37%	**0.2%**
24	99.33%	98.9%	98.8%	99.66%	91.87%	99.74%	96.97%	97.64%	**0.2%**
12	98.71%	98.8%	98.6%	99.58%	78.06%	99.68%	90.54%	92.14%	0.4%
6	98.08%	98.3%	98.1%	99.43%	65.4%	99.59%	84.35%	86.8%	0.6%
4	97.74%	98.1%	97.8%	99.28%	60.29%	99.53%	82.47%	85.08%	0.6%

TABLE 4
Performance Values for the LSTM Models Obtained by Varying *Prediction Window (q)* and TW Size on the Backblaze Data-Set

q [day]	*TW SIZE [day]*	*Accuracy*	*Precision*	*Recall*	ACC_G	ACC_F	ACC_G^{TOL}	ACC_F^{TOL}	*FDR*	*FAR*
15	5	95.88%	96.90%	95.10%	97.28%	66.56%	97.89%	98.08%	75.53%	2.82%
15	7	95.81%	97.10%	96.00%	97.02%	70.27%	97.93%	**98.45%**	79.34%	2.70%
30	5	94.54%	96.50%	94.60%	96.38%	56.07%	97.68%	88.30%	76.03%	2.73%
30	7	93.93%	96.80%	94.40%	95.59%	59.15%	97.07%	89.37%	80.70%	3.29%
30	10	95.25%	97.40%	96.10%	96.84%	61.84%	97.59%	91.35%	85.48%	2.73%
45	5	94.45%	96.70%	94.93%	95.95%	66.16%	97.80%	90.67%	78.30%	2.50%
45	7	95.82%	97.00%	95.85%	97.28%	68.34%	98.12%	89.37%	77.75%	2.17%
45	10	96.56%	97.72%	96.82%	97.71%	75.08%	98.36%	93.30%	84.18%	1.83%
45	14	**98.45%**	**98.33%**	**98.34%**	**99.21%**	**84.49%**	**99.40%**	96.65%	**91.48%**	**0.72%**

defined in this paper is a multiclass classification problem with multivariate input variables.

The performance of our approach is first evaluated in terms of accuracy, precision, and recall. Since the distinction between good and failed hard drives is preserved in the labelling of the data-set, we express the results in term of accuracy on good sequences (ACC_G) and accuracy on failed sequences (ACC_F) — respectively, the fraction of sequences correctly classified as *Good*, and the fraction of sequence classified as the health levels suggested by the regression trees. We also consider the evaluation criteria introduced in [4], and measure the accuracy of classifying good and failed sequences for a tolerance of misclassification up to one health level (ACC_G^{TOL} and ACC_F^{TOL}).

Finally, we evaluate performance in terms of failure prediction, by assessing *failure detection rate* (FDR) and *false alarm rate* (FAR) for each model. This is done by considering the levels *Good*, *Very Fair*, and *Fair* as *Hard drive good statuses*; and the levels *Soft Warning*, *Warning*, *Alert*, and *Red Alert* as *Hard drive failed statuses* (see Figs. 4 and 7). Intuitively, FDR is the fraction of failed sequences that are correctly classified as failed, while FAR is the fraction of good sequences that are incorrectly classified as failed.

7 Results

In this paper we proposed a methodology to perform hard drive health status assessment exploiting the temporal dependencies of SMART attributes. In order to asses the effectiveness of the proposal, this section reports the performance of our methodology, and a comparison with several state-of-art approaches.

First, we report the results for both sequence dependent and sequence independent approaches. In particular, Tables 3 and 4 show results of the LSTM based approach on the Baidu and Backblaze data-sets, respectively. Performance is reported for different sizes of the time window (TW) used in the sequence extraction step. We explored time window sizes from 4 to 48 hours for the Baidu data-set, and from 5 to 15 days for the Backblaze data-set. For the latter, we considered a *prediction window (q)* varying from 15 to 45 days. As expected given the ability of LSTMs to learn long-distance dependencies, the best results are obtained with time windows of 48 hours and 15 days for the Baidu and Backblaze data-sets, respectively.

Tables 5 and 6 report results for the sequence independent models. More in details, such models take hourly samples as input rather than sequences. The best results in terms of accuracy on failed sequences are obtained with RF for the Baidu data-set, and MNN for the Backblaze data-set. Results show that a sequence dependent approach provides higher performance than a sequence independent methodology, since the former is able to capture the SMART attribute temporal

TABLE 5
Results of Sequence Independent Models on the Baidu Data-Set

Model	*Accuracy*	ACC_G	ACC_F	ACC_G^{TOL}	ACC_F^{TOL}	*FDR*	*FAR*
CT	97.01%	97.01%	58.94%	99.09%	**85.77%**	84.16%	1.00%
RF	**98.13%**	98.13%	**59.44%**	**99.82%**	85.65%	**85.36%**	**0.40%**
MNN	96.24%	**98.57%**	38.99%	99.14%	69.59%	73.03%	1.20%

TABLE 6
Results of Sequence Independent Models on the Backblaze Data-Set

Model	*Accuracy*	ACC_G	ACC_F	ACC_G^{TOL}	ACC_F^{TOL}	*FDR*	*FAR*
CT	83.80%	83.87%	56.31%	95.63%	88.46%	63.58%	4.69%
RF	85.77%	85.77%	**71.75%**	93.68%	**93.82%**	**80.66%**	6.49%
MNN	**96.17%**	**99.15%**	39.78%	**99.88%**	69.20%	85.75%	**0.95%**

TABLE 7
Results of Best Model on the Baidu Data-Set Detailed by Each Class

Metric	*Good*	*Very Fair*	*Fair*	*Soft Warning*	*Warning*	*Alert*	*Red Alert*
Accuracy	99.49%	75.10%	63.17%	41.39%	72.60%	47.44%	61.88%
Precision	100.00%	58.40%	50.90%	57.10%	46.60%	59.20%	60.10%
Recall	99.30%	75.06%	63.20%	41.40%	72.40%	47.30%	61.90%

TABLE 8
Results of Best Model on the Backblaze Data-Set Detailed by Each Class

Metric	*Good*	*Very Fair*	*Warning*	*Alert*
Accuracy	99.21%	87.80%	78.10%	84.42%
Precision	99.90%	69.40%	64.70%	73.10%
Recall	98.80%	87.80%	78.10%	84.40%

dependencies. For completeness, Tables 7 and 8 report the performance of our best models detailed by each class.

In order to evaluate the effect of the automatic health degree definition step (as detailed in Section 4.1), Table 11 compares the performance of the model using automatically *or* manually selected health levels. For the manual set-up, hard drive health levels were split only considering the features *Hour to failure* and *Day to failure* — respectively for the Baidu and Backblaze data-set. Specifically, we define weekly (seven-day long) intervals. The only exception being the first interval, which is defined as relative to three days before failure. This comparison clearly highlights the effectiveness of automatically detected degree levels, as the automatic approach consistently outperforms the manual split the variety of evaluation metrics considered.

Furthermore, we evaluate how the balancing method affects performance, by comparing our approach with respect to the methods in [23] and [24]. In particular, Botezatu *et al.* [23] selected a representative subset of healthy disks by means of a K-means clustering algorithm, while Xu *et al.* [24] applied an over-sampling technique (SMOTE) to balance the minority classes. Tables 9 and 10 report the results obtained by varying the balancing method, and show that the best results on both data-sets are obtained using our method.

Finally, we performed a comparison between our methodology and some other proposals in the literature, which had also been tested on the SMART data-set. Tables 12, 13, 14 and 15 compare our best results on the Baidu and Backblaze data-sets with different approaches for hard drive health status assessment and hard drive failure prediction tasks. Results for the first experiment are shown in Table 12. We compared the performance of our approach on the Baidu data-set against a method based on Recurrent Neural Networks (RNN) (Xu *et al.* [4]), a model based on a Multiclass Neural Network (Mutliclass NN), and one based on Conditional Random Fields (CRFs) for hard drive health status assessment.

TABLE 9
Results Obtained by Varying Different Balancing Methods on the Backblaze Data-Set

Model	*Accuracy*	ACC_G	ACC_F	ACC_G^{TOL}	ACC_F^{TOL}	*FDR*	*FAR*
our approach	**98.45%**	99.21%	**84.49%**	99.40%	96.65%	**91.48%**	**0.72%**
K-Means [23]	93.90%	99.70%	61.76%	99.20%	88.10%	74.47%	2.30%
Smote [24]	97.37%	**99.86%**	51.87%	**99.88%**	83.68%	58.43%	0.80%

TABLE 10
Results Obtained by Varying Different Balancing Methods on the Baidu Data-Set

Model	*Accuracy*	ACC_G	ACC_F	ACC_G^{TOL}	ACC_F^{TOL}	*FDR*	*FAR*
our approach	**99.80%**	99.83%	**93.17%**	99.89%	**98.31%**	**98.2%**	**0.2%**
K-Means [23]	92.32%	99.91%	38.26%	99.95%	68.36%	68.77%	1.09%
Smote [24]	98.03%	**99.95%**	51.55%	**99.97%**	77.19%	76.40%	0.29%

TABLE 11
Results Obtained by Varying Different Methods to Define Hard Drive Health Levels

Dataset	*Method*	*ACC*	ACC_G	ACC_F	ACC_G^{TOL}	ACC_F^{TOL}	*FDR*	*FAR*
Backblaze	manual	97.54%	98.78%	75.26%	99.06%	93.31%	90.63%	0.89%
Backblaze	automatic	**98.45%**	**99.21%**	**84.49%**	**99.40%**	**96.65%**	**91.48%**	**0.72%**
Baidu	manual	99.15%	98.95%	92.38%	99.29%	97.74%	97.82%	0.37%
Baidu	automatic	**99.80%**	**99.83%**	**93.17%**	**99.89%**	**98.31%**	**98.20%**	**0.20%**

TABLE 12
Comparison of Our Best Model (LSTM - 48h) on the Baidu Data-Set With Previously Proposed Models on the Hard Drive Health Status Assessment Task

Author	Methods	ACC_G	ACC_F	ACC_G^{TOL}	ACC_F^{TOL}
Xu *et al.* [4]	Multiclass NN	99.19%	16.01%	99.40%	43.34%
Xu *et al.* [4]	CRF	99.57%	28.51%	99.59%	61.30%
Xu *et al.* [4]	RNN	99.73%	41.05%	**99.93%**	64.86%
Our Approach	LSTM	**99.83%**	**93.17%**	99.89%	**98.31%**

In Table 13, we consider once more the models in Xu *et al.* [4], which were adapted to the hard drive failure prediction task by implementing a voting rule mapping different health levels to two separate classes. Then, we consider the models in *Li et al.* [11], Zhu *et al.* [12] and Shen *et al.* [3] — respectively, a Classification Tree (CT) model, a Backpropagation (BPNN) and a Random Forest (RF) model.

TABLE 13
Comparison of Our Best Model (LSTM - 48h) on the Baidu Data-Set With Previously Proposed Models on the Hard Drive Failure Prediction Task

Author	Methods	*FDR*	*FAR*
Xu *et al.*[4]	Multiclass NN	83.21%	0.60%
Xu *et al.*[4]	CRF	85.50%	0.22%
Xu *et al.*[4]	RNN	87.79%	**0.004%**
Li *et al.* [11]	CT	95.49%	0.09%
Zhu *et al.* [12]	BP NN	94.62%	0.48%
Shen *et al.* [3]	RF	97.67%	0.017%
Our Approach	LSTM	**98.20%**	0.20%

TABLE 14
Comparison of Our Best Model (LSTM - $TW = 14$ Days and $q = 45$ days) on the Backblaze Data-Set With Previously Proposed Models on the Hard Drive Health Status Assessment Task

Author	Methods	Accuracy	Precision	Recall
Zhang *et al.*[9]	LPAT+All	92.6%	89.30%	88.70%
Sun *et al.*[25]	TCNN	—	75.00%	67.00%
Basak *et al.*[7]	LSTM	—	84.35%	72.00%
Our Approach	LSTM	**98.45%**	**98.33%**	**98.34%**

TABLE 15
Comparison of Our Best Model (LSTM - $TW = 14$ days and $q = 45$ days) on the Backblaze Data-Set With Previously Proposed Models on the Hard Drive Failure Prediction Task

Author	Methods	*FDR*	*FAR*
Shen *et al.*[3]	RF	94.89%	0.44%
Xiao *et al.*[20]	ORF	98.08%	0.66%
Our Approach	LSTM	**98.20%**	**0.20%**

Lastly, Tables 14 and 15 compare our best result on the Backblaze data-set with other state-of-the-art methods in the literature: Zhang *et al.* [9], a method based on adversarial training and layerwise perturbation (LPAT); Sun *et al.* [25], a temporal convolutional neural network for failure prediction; Basak *et al.* [7], an LSTM-based prediction model for RUL estimation; Shen *et al.* [3] and Xiao *et al.* [20], a prediction model based on part-voting Random Forest and Online Random Forest.

To summarize, our proposal outperforms all these models in terms of accuracy on failed sequences, FDR, and FAR both for hard drive health status assessment and hard drive failure prediction tasks. Importantly, experimental results demonstrate that our approach is feasible for HDD health status assessment task due to the pre-processing phase and the definition of a specific model (LSTM) relying on temporal sequence. Crucially, by showing how our model outperforms exiting methods based on LSTMs and CNNs (Table 14), these comparisons highlight the essential contribution of our approach.

8 Conclusion

In the past decades, being able to predict the health level of HDD timely and accurately has started playing a fundamental role in the administration of large data centers, as optimizing maintenance strategies has obvious impacts on overhead costs.

This paper proposes a methodology for predicting hard drives' health level which combines machine learning prediction techniques based on LSTMs, with an automatic approach for hard drive health status definition.

LSTMs are interesting in the context of HDD failure prediction, as they are able to take advantage of the highly sequential nature of the information available to the model. To explore the effectiveness of this idea, we investigated how extracting sequences over time windows (TW) of different sizes affects the performance of the LSTM model. In line with the ability of LSTMs to learn long-distance dependencies, the best results were obtained with 48-hour time windows size and 14-day time windows size for Baidu and Backblaze datasets respectively. Furthermore, we showed that LSTM based models outperform sequence independent models in classifying sequences belonging to hard drives that are going to fail. As can be seen from the summary Tables 3, 4, 5 and 6 the performance gap between models on healthy samples or sequences is small. We interpret this as evidence of the fact that it is not difficult for a classifier to identify good sequences, but it is hard to identify disks at risk of failure. In such cases, the LSTM models strongly outperforms the sequence independent models.

We achieve state-of-the-art results on a hard drive health status assessment task (Tables 12 and 14) over the SMART data-set, and competitive results in terms of FDR and FAR scores (Tables 13 and 15). Moreover, as shown in Tables 13 and 15, our approach outperforms the others in term of FDR. Although our model does not result in the lowest FAR value (Table 13), it keeps it to a reasonably low value. Importantly, the advantage of our solution compared to that introduced in [4] lies specifically in the implementation of a step for the automatic definition of health levels.

Notably, LSTMs have been already used in the past in predictive maintenance tasks.

The fundamental contribution of this paper though, is in the way LSTMs are coupled with a technique to address the unbalanced nature of the training data, and the data-driven definition of health degree levels. Traditionally, HDD have been classified based on fixed health levels determined by domain experts. In this sense, optimal maintenance strategies would benefit from as detailed a prediction as possible, while maintaining high accuracy. However, the complex nature of HDD health status makes this technique less effective over real world data, thus requiring trade-offs between detailed and timely predictions, and prediction accuracy. By providing an automatic methodology to detect ranges on the base of each disk's behavior, our approach shows more flexibility to the varied nature of the underlying data, thus outperforming a variety of alternative models.

Comparisons with alternative methods also based on LSTMs support the effectiveness of the particular approach taken in this paper, and shows the advantages (in terms of accuracy of the prediction task) of enriching LSTMs with data-driven, flexible identification of HDD health levels and health degree settings (Tables 14 and 15).

Additionally, our model automatically extracted degree levels were assessed over a range of prediction windows (15 to 45 days) compatible with what already present in the literature, thus supporting the reliability of the approach in real-word scenarios.

Based on such results, we argue that the health level prediction task should be preferred to other forms of evaluations when considering these kind of predictive models, as its results provide the most insightful information towards hard drive maintenance. Future work should be focused on exploring the best way data center's technicians and manager can leverage the finer-grained predictions provided by our approach to optimize long-term maintenance. Furthermore, we should conduct an extensive, detailed investigation of different health degree settings, evaluating the trade-offs of incorporating constraints from real-world applications into the automated processes discussed in this paper so to further refine health degree definition strategies.

REFERENCES

[1] G. Hamerly *et al.*, "Bayesian approaches to failure prediction for disk drives," in *Proc. Int. Conf. Mach. Learning*, 2001, vol. 1, pp. 202–209.

[2] C.-J. Su and S.-F. Huang, "Real-time big data analytics for hard disk drive predictive maintenance," *Comput. Elect. Eng.*, vol. 71, pp. 93–101, 2018.

[3] J. Shen, J. Wan, S.-J. Lim, and L. Yu, "Random-forest-based failure prediction for hard disk drives," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 11, pp. 1–15, 2018.

[4] C. Xu, G. Wang, X. Liu, D. Guo, and T.-Y. Liu, "Health status assessment and failure prediction for hard drives with recurrent neural networks," *IEEE Trans. Comput.*, vol. 65, no. 11, pp. 3502–3508, Nov. 2016.

[5] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, "Machine learning methods for predicting failures in hard drives: A multiple-instance application," *J. Mach. Learn. Res.*, vol. 6, no. May, pp. 783–816, 2005.

[6] N. Aussel, S. Jaulin, G. Gandon, Y. Petetin, E. Fazli, and S. Chabridon, "Predictive models of hard drive failures based on operational data," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl.*, 2017, pp. 619–625.

[7] S. Basak, S. Sengupta, and A. Dubey, "Mechanisms for integrated feature normalization and remaining useful life estimation using lstms applied to hard-disks," in *Proc. IEEE Int. Conf. Smart Comput.*, 2019, pp. 208–216.

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

[9] J. Zhang, J. Wang, L. He, Z. Li, and S. Y. Philip, "Layerwise perturbation-based adversarial training for hard drive health degree prediction," in *Proc. IEEE Int. Conf. Data Mining*, 2018, pp. 1428–1433.

[10] Z. Kong, Y. Cui, Z. Xia, and H. Lv, "Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics," *Appl. Sci.*, vol. 9, no. 19, 2019, Art. no. 4156.

[11] J. Li *et al.*, "Hard drive failure prediction using classification and regression trees," in *Proc. 44th Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw.*, 2014, pp. 383–394.

[12] B. Zhu, G. Wang, X. Liu, D. Hu, S. Lin, and J. Ma, "Proactive drive failure prediction for large scale storage systems," in *Proc. IEEE 29th Symp. Mass Storage Syst. Technol.*, 2013, pp. 1–5.

[13] T. Coughlin, "MAMR hard disk drives enable future data centers," 2017. Accessed: Jun. 24, 2019. [Online]. Available: https://www.forbes.com/sites/tomcoughlin/2017/10/11/mamr-hard-disk-drives-enable-future-data-centers/#77f5702e2054

[14] A. Klein, "Backblaze Hard Drive stats," 2019. Accessed: Oct. 4, 2019. [Online]. Available: https://www.backblaze.com/blog/backblaze-hard-drive-stats-q1-2019/

[15] R. Bauer, "HDD vs SSD: What does the future for storage hold?," 2018. Accessed: Oct. 4, 2019. [Online]. Available: https://www.backblaze.com/blog/ssd-vs-hdd-future-of-storage/

[16] A. Klein, "The shocking truth: Managing for Hard Drive failure and data corruption," 2019. Accessed: Oct. 4, 2019. [Online]. Available: https://www.backblaze.com/blog/managing-for-hard-drive-failures-data-co rruption/

[17] G. A. Gibson and D. A. Patterson, "Designing disk arrays for high data reliability," *J. Parallel Distrib. Comput.*, vol. 17, no. 1–2, pp. 4–27, 1993.

[18] Z. Qiao, J. Hochstetler, S. Liang, S. Fu, H.-B. Chen, and B. Settlemyer, "Developing cost-effective data rescue schemes to tackle disk failures in data centers," in *Proc. Int. Conf. Big Data*, 2018, pp. 194–208.

[19] C.-J. Su and Y. Li, "Recurrent neural network based real-time failure detection of storage devices," *Microsystem Technologies*, 2019. [Online]. Available: https://doi.org/10.1007/s00542-019-04454-8

[20] J. Xiao, Z. Xiong, S. Wu, Y. Yi, H. Jin, and K. Hu, "Disk failure prediction in data centers via online learning," in *Proc. 47th ACM Int. Conf. Parallel Process.*, 2018, Art. no. 35.

[21] Y. Wang, E. W. Ma, T. W. Chow, and K.-L. Tsui, "A two-step parametric method for failure prediction in hard disk drives," *IEEE Trans. Ind. Inf.*, vol. 10, no. 1, pp. 419–430, Feb. 2014.

[22] P. Anantharaman, M. Qiao, and D. Jadav, "Large scale predictive analytics for hard disk remaining useful life estimation," in *Proc. IEEE Int. Congr. Big Data*, 2018, pp. 251–254.

[23] M. M. Botezatu, I. Giurgiu, J. Bogojeska, and D. Wiesmann, "Predicting disk replacement towards reliable data centers," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 39–48.

[24] Y. Xu *et al.*, "Improving service availability of cloud systems by predicting disk error," in *Proc. {USENIX} Annu. Techn. Conf.*, 2018, pp. 481–494.

[25] X. Sun *et al.*, "System-level hardware failure prediction using deep learning," in *Proc. 56th ACM/IEEE Des. Autom. Conf.*, 2019, pp. 1–6.

[26] H. Shao and B.-H. Soong, "Traffic flow prediction with long short-term memory networks (LSTMs)," in *Proc. IEEE Region 10 Conf.*, 2016, pp. 2986–2989.

[27] M. A. Zaytar and C. El Amrani, "Sequence to sequence weather forecasting with iong short-term memory recurrent neural networks," *Int. J. Comput. Appl.*, vol. 143, no. 11, pp. 7–11, 2016.

[28] T. Linzen, E. Dupoux, and Y. Goldberg, "Assessing the ability of LSTMs to learn syntax-sensitive dependencies," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 521–535, 2016.

[29] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. Shelter Island, NY, USA: Manning Publications, 2015.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

Aniello De Santo received the bachelor's and master's degrees in computer science and engineering from the University of Pavia, and the PhD degree in Linguistics from Stony Brook University, in 2020. He is an assistant professor with the Department of Linguistics at the University of Utah. His main research interests focus on formal language-theoretical approaches to the complexity of natural languages, rich grammar formalisms (MGs, TAG), and cognitively plausible models of human sentence processing.

Antonio Galli received the bachelor's and master's degree cum laude in computer science and engineering from the University of Naples "Federico II", in 2016 and 2019, respectively. He is currently working toward the PhD degree in technology, innovation and management in the University of Bergamo and the University of Naples "Federico II". His main research interests are in the area of deep learning and big data analytics.

Michela Gravina received the bachelor's (cum laude) and master's (cum laude) degrees in computer science and engineering from the University of Naples "Federico II", in 2016, 2019, respectively. She is currently working toward the PhD degree in information and communication technology for health in the University of Naples "Federico II". Her main research interests are in the area of deep learning and big data analytics.

Vincenzo Moscato is currently an associate professor of Database and Information Systems with the Department of Electrical Engineering and Information Technologies of University of Naples "Federico II". His current research interests include the area of multimedia, knowledge management, and Big Data analytics. He was involved in several international, national, and local research projects and at present is an author of more than one hundred publications in international journal and conference proceedings.

Giancarlo Sperlí received the bachelor's and master's degrees in computer science and engineering, both from the University of Naples "Federico II", respectively, and the PhD degree in information technology and electrical engineering from the same university, in 2018. He is a researcher fellow with the University of Naples "Federico II". His main research interests include the area of cybersecurity, semantic analysis of multimedia data, and social networks analysis.