Dynamic Sample Weighting to Predict the Remaining Useful Life of Hard Disk Drives

Gabriel S. Felix, Francisco F. Pereira, Francisco D. Praciano, João P. Gomes, Javam C. Machado

LSBD - Universidade Federal do Ceará, Brazil
{gabriel.felix, lucas.falcao, daniel.praciano, joao.pordeus, javam.machado}@lsbd.ufc.br

Abstract. Hard Disk Drives (HDDs) are widely used for data storage in various applications. However, their failure can result in significant data loss and system downtime. Therefore, accurate prediction of the remaining useful life (RUL) of HDDs is crucial for proactive maintenance and data backup strategies. In this paper, we propose a novel approach to predict the RUL of HDDs using Long Short-Term Memory (LSTM) networks and incorporating weighted loss functions. The proposed model leverages the Self-Monitoring, Analysis, and Reporting Technology (SMART) features of HDDs, which provide valuable information about the health of the drive. We evaluated two weighting approaches that improve the general performance and enhance predictions within a given timeframe. Our experiments showed that the models outperformed traditional methods in terms of Mean Squared Error (MSE) at given time intervals.

CCS Concepts: • **Computing methodologies** → **Machine learning algorithms**.

Keywords: HDD, RUL, Failure prediction, Deep Learning, Sample Weighting

1. INTRODUCTION

The degradation of components within Hard Disk Drives (HDDs) is widely recognized as a potential cause of severe data loss. Consequently, there is a pressing need to identify and predict the deterioration of HDDs. To address this challenge, manufacturers introduced the Self-Monitoring, Analysis, and Reporting Technology (SMART) system. SMART continuously monitors various disk parameters and compares them against predefined thresholds. Despite its widespread use, the failure detection rate of SMART is typically low, ranging from 3% to 10% [Murray et al. 2005].

Recently, researchers have made substantial efforts to develop more dependable methods for predicting HDD failures. Many of these studies involve integrating machine learning techniques with the SMART attributes. Notably, [Murray et al. 2005] conducted one of the pioneering works in this field, exploring multiple machine learning algorithms for such task. [Chaves et al. 2016] employed Bayesian Networks, while [Lima et al. 2021] achieved some of the most promising results using Long Short-Term Memory (LSTM) models. Other deep learning-based models achieved remarkable results that are reported in [Cahyadi and Forshaw 2021; Pereira et al. 2022; Hu et al. 2020].

Despite extensive research conducted and the results of deep learning models, previous developments in the field have treated all Remaining Useful Life (RUL) predictions as equally important during the training phase. This means that both long-term and short-term predictions have had the same impact on the loss function of the model. As a result, the final network has not been specifically designed to prioritize either long or short predictions, even though such a characteristic may be desirable. However, in real-world applications, the ability to predict failures within a specific timeframe before they occur is crucial for effective maintenance planning. Therefore, incorporating this feature into the model becomes essential for enabling adequate maintenance strategies.

To incorporate such feature in neural network models, we evaluate the use of weighted loss function to LSTMs. In this work, we introduce two different weighting schemes for the cost functions. The first scheme assigns weights to training samples based on their distance to the end of life of the HDD. By emphasizing the importance of accurately predicting samples far/close to failure, our model aims to capture the critical period leading up to HDD failure.

The second weighting scheme assigns weights to each sample based on its prediction error during a training epoch. This approach does not focus on improving the performance of LSTM at any timeframe before the failure but to alleviate the impact of very poor predictions, allowing the model to focus on improving its overall performance by reducing the influence of outliers.

By using one of these two weighting schemes and by combining them, our proposals may address the importance of accurate predictions near the end/beginning of life and the need to mitigate the impact of highly erroneous predictions. This comprehensive approach aims to improve the final prediction accuracy and enhance the reliability of HDD failure prediction.

Experimental evaluations conducted on a large dataset of HDDs demonstrate the effectiveness of our proposed models. Compared to existing approaches, our models achieve superior prediction accuracy when analyzing different time intervals.

2. BACKGROUND

2.1 HDD Failure Prediction

As aforementioned, the usual method for diagnostic monitoring in hard disk drives is through the SMART technology [Ottem and Plummer 1995]. The HDD manufacturer defines a set of attributes from sensors to error counters, such as temperature sensor, flying high sensor, read error rate, reallocated sectors count, etc. The manufacturer also defines a threshold for the attributes that, once it's reached, indicates an imminent failure.

Also, studies such as [Murray et al. 2005] and [Pinheiro et al. 2007] concluded that predictions based on those thresholds have low accuracy, and that SMART data has limited usefulness in anticipating disk failure. Therefore, the two main issues with the SMART threshold approach are its limited utility to detect faulty disks and the fact that it only detects near-term failures.

To help to solve those issues for failure prediction, studies have been dealing with the task in different ways. One of them is to model the problem as a classification task, where day intervals from the Remaining Useful Life (RUL) are seen as different health levels (or classes) [Lima et al. 2021]. Another way is to treat the problem as a regression on the RUL directly [Lima et al. 2018]. In this work we will follow the second approach, i.e. for each hard drive, on a given day, the model will predict the remaining days until a failure.

2.2 Long Short-Term Memory

The default Recurrent Neural Networks (RNNs) cannot capture long dependencies in the context. Such a fact encouraged the design of Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber 1997], which works by adding gate mechanisms that control the entrance and the exit of information in the cell state. These gates are associated with internal memory to RNN cells, controlling the flow of information from the input and the previous states. The LSTM cell is shown in Fig. 1 (adapted from [Olah 2015]).

The LSTM unit has the forget, input, and output gates to control the information flow. The forget gate controls how much information the unit accepts from the input and the last state. On the other hand, the input state regulates how much information is added to the current state cell. The output gate determines how much information can be output from the current cell state. The equations below

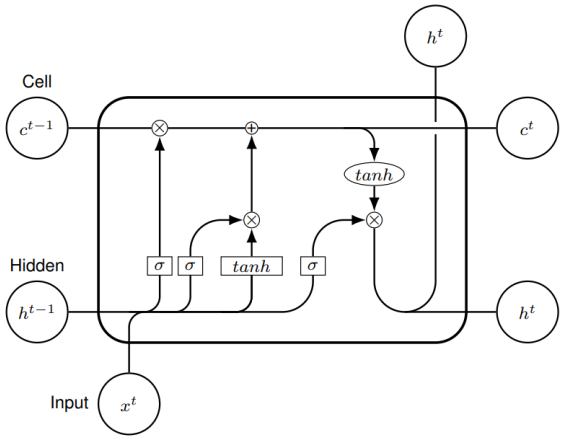

Fig. 1: LSTM cell (adapted from [Olah 2015])

define how to calculate all these gates, cell states, and hidden states for the forward pass of an LSTM layer. In these equations, t denotes the processed index within a sequence x. The symbol $\odot$ represents the Hadamard product. W and U are both the recurrent and input matrices, respectively, with a subscript indicating the associated gate. The parameter b is the bias term and the c and h are the cell and hidden state, respectively.

$$\widetilde{c}_t = tanh(W_c h_{t-1} + U_c x_t + b_c) \quad \text{candidate state} \quad (1)$$
$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \quad \text{input gate} \quad (2)$$
$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \quad \text{forget gate} \quad (3)$$
$$c_t = i^t \odot \widetilde{c}^t + f^t \odot c^{t-1} \quad \text{cell state} \quad (4)$$
$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \quad \text{output gate} \quad (5)$$
$$h_t = o_t \odot tanh(c_t) \quad \text{output} \quad (6)$$

3. METHODOLOGY

3.1 Sample weighting methods

Sample weighting in training is a technique usually employed to correct biases, such as solving data imbalance. It does that by giving more importance to specific samples defined by a criterion.

In the same way, we can use sample weighting to correct bias in data, we also can use it to force bias in the training phase. In this work, we are focusing on how to use sample weighting to improve the Remaining Useful Life prediction of hard disk drives. When no sample weighting is employed, the same importance is given to every sample of the data. We will call this baseline model as **equal** weighting.

Since we are dealing with the problem of predicting the failure of a device, it is reasonable to consider the closeness to the failure day as an important aspect to model. Thereby, we propose a way to emphasize this by using sample weights.

Therefore, we are proposing to define sample weight functions based on the ground truth Remaining Useful Life. Since the days close to the failure point are more critical, it is natural to give more weight the closer the prediction is to them. Following this, the first RUL based sample weighting we are proposing is a linear function that maps the ground truth RUL to a weight range. We can describe

it as a function that maps linearly the domain $[1, 360]$ to $[W_{max}, W_{min}]$, where $W_{min} < W_{max}$ are hyperparameters. Their difference measures how much importance we should give to the last day. In this work, the predictive model can make predictions up to 360 days, which is why the domain is limited. However, it can be increased for predictive models with longer range. We call this the **Dynamic Sample Weighting - Decreasing (DSW-D)** weighting method.

Although giving more weight to the last days of life of the device is reasonable, one may want to focus on the opposite, increasing the prediction accuracy in a timeframe way before the last day of the failure. This approach may be useful to avoid false alarms with more trustworthy predictions in the long term. In a similar way to the decreasing weighting, we can describe the new method as a function that maps linearly the domain $[1, 360]$ to $[W_{min}, W_{max}]$, where $W_{min} < W_{max}$ are hyperparameters. Their difference measures how much importance we should give to the longest prediction the model can make. We call this the **Dynamic Sample Weighting - Increasing (DSW-I)** weighting method.

Also, the model predictions can be affected negatively by the presence of very poor predictions in the training phase. For this reason, we are also proposing an approach to consider it by changing the sample weights. We are proposing to use the absolute error of each sample as part of its weights. This is done by combining the current error with the historical weight given to each sample in previous epochs through an affine combination. The equation that defines this method is shown in 7, where i is the current epoch, $error[i]$ is the absolute error of the epoch i and $0 < \alpha < 1$ is a hyperparameter that defines how much of the error of the current epoch should be considered, similar to the smoothing factor in the exponential smoothing formula. Since the weights from one epoch to the next are being combined, to avoid large differences in those weights, a linear normalization is performed to set the weight values to the range $[W_{min}, W_{max}]$, where $W_{min} < W_{max}$. We call this the **Dynamic Sample Weighting - historical Error based (DSW-E)** weighting method.

$$weights[i] = \alpha \times error[i] + (1 - \alpha) \times weights[i - 1] \tag{7}$$

Also, since the sample weighting methods are simple multiplications to the sample errors, more than one weighting scheme can be applied to them. Therefore, we are also proposing to combine these methods by applying the historical error based and the increasing (**DSW-E+I method**), and by applying the historical error based and the decreasing (**DSW-E+D method**).

3.2 Metrics

To properly assess the model's predictions, the used metrics are an important decision. Since it is a regression task, for the general evaluation of the results, the Mean Squared Error (MSE) is an appropriate metric.

However, since we are studying the impact of sample weights in different time frames of the life of HDDs, different metrics should be used to take it into account. For that reason, we are also using the MSE at the first and the last useful 30 days of life of the HDDs as two new separate metrics. With this, we should be able to compare the impact of sample weighting in far and close-to-failure scenarios.

It is important to state that, even though this study is about sample weighting in training, these metrics are the regular MSE with no weighting to the samples.

4. EXPERIMENTS AND RESULTS

4.1 Dataset

To evaluate the performance of the methods, we conducted a set of experiments using a public dataset provided by the Backblaze Company [Backblaze 2023]. This dataset contains daily reports of SMART attributes of several HDDs, ranging from April/2013 until December/2022. These observations contain the serial number, model, SMART attributes, and a label indicating if the HDD has failed or if it presented some indication that will fail.

For this study, we selected the HDD manufacturer model with the largest amount of data, specifically the Seagate ST4000DM000 model, and selected only serials that failed. We limited our analysis to the last 360 days of HDD life and excluded serials whose daily observations were interrupted without a faulty label or disks with data measurements after being labeled faulty. After these processes, our dataset contains 1,631,802 daily observations divided into 4,936 serials.

To improve the results of the model, we use only a limited number of the SMART attributes collected in the dataset, chosen by sequential feature selection with the linear regression estimator from Scikit-Learn 1.0.2 library. These features are described in Table I.

Attribute ID	**Attribute Name**
SMART 9	Power-On Hours
SMART 10	Spin Retry Count
SMART 12	Power Cycle Count
SMART 184	End-to-End Error
SMART 187	Reported Uncorrectable Errors
SMART 190	Temperature Difference
SMART 193	Load Cycle Count
SMART 194	Temperature
SMART 197	Current Pending Sector Count
SMART 240	Head Flying Hours
SMART 241	Total LBAs Written
SMART 242	Total LBAs Read

Table I: SMART attributes used as features to the model

4.2 Experimental procedure

In this study, we use a model to predict, for each day, the remaining useful life (RUL) of HDDs in days, modeling it as a regression task. Every disk data is a time series and we used 60% of the disks for the train, 20% of the disks for the validation, and 20% of the disks for the test.

The analysis of the preprocessed dataset showed us that the vast majority of samples, but not all, have 360 days of observations. To balance our dataset, we partitioned each training sample randomly, into up to three parts, containing each part at a minimum of 30 days, according to its size. Every 120 days we add one partition, and combine them to form new samples, as shown in Fig. 2. After this process, only the amount of train samples changed, so we have 16,199 train samples, 976 validation samples, and 1032 test samples. The model uses the validation set as a criterion for early stopping.

To evaluate the performance of the methods under discussion, we used a model composed of an LSTM, followed by a Dense layer, a Dropout layer, and another Dense layer. The Min-max normalization is applied to the model output. The model flowchart is shown in Fig. 3, and its hyperparameters can be consulted in Table II.

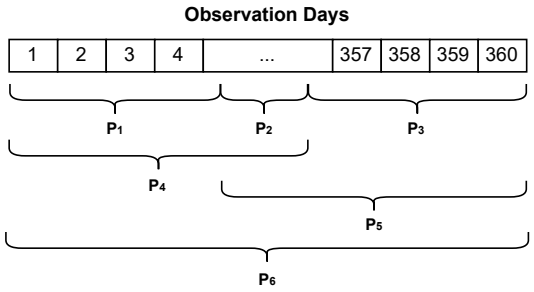

Fig. 2: Days partition for the training dataset.

Hyperparameter	Value
Epochs	5,000
Learning Rate	1e-3
Batch Size	1,024
Early Sttoping Patience	600
Optimizer	Adam
LSTM Cell Dimension	128
Dense 1 Output Dimension	64
Dense 2 Output Dimension	1
Dense 1 Activation	Relu
Dense 2 Activation	
Dense 1 L1 Penalty	1e-5
Dense 1 L2 Penalty	
Dense 2 L1 Penalty	
Dense 2 L2 Penalty	
Dropout Rate	1e-1

Table II: Predictive model hyperparameters.

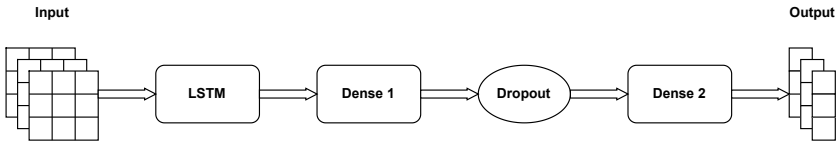

Fig. 3: Predictive model. The input is the time series for each HDD and the output is the RUL prediction.

Also, linear normalization was applied to all proposed methods. The W_{min} and W_{max} hyperparameters of methods are set, respectively, to 1 and 10. In the historical error-based method, the smoothing factor (α) was set to 0.3. Each method was run 3 times using TensorFlow 2.8.0.

4.3 Results

	Test MSE	First 30 Days MSE	Last 30 Days MSE
Equal (baseline)	0.01720	0.05552	0.02374
DSW-Increasing	0.01180	0.03753	0.03470
DSW-Decreasing	0.03794	0.15166	**0.01796**
DSW-E	0.01159	0.03724	0.02561
DSW-E+Increasing	**0.00950**	**0.03046**	0.02932
DSW-E+Decreasing	0.02359	0.09934	0.01926

Table III: Results for the test dataset.

Table III presents the performance of each sample weighting method for each metric. As aforementioned, each method was run 3 times and the table shows their average. As it can be noticed, for the standard MSE metric, the historical error-based + increasing (DSW-E+I) has the best result, followed by the historical error-based (DSW-E) employed alone. This supports the idea that the DSW-E method can improve the predictions by focusing on large prediction errors, improving the overall performance.

On the other hand, for the first 30 useful days MSE metric, again, the historical error based + increasing method has the best result, followed by the historical based error employed alone. It shows us that, in fact, the Increasing (DSW-I) method works to obtain more accurate predictions far to failure of HDDs, and when combined with the DSW-E improves substantially the accuracy of model predictions.

For the last 30 days MSE metric, the Decreasing (DSW-D) method is the best, followed by the historical error based + decreasing (DSW-E+D). Although the difference between those two best methods is small, it may indicate that the close-to-failure time frame is difficult to predict, so focusing only on the last days (DSW-D method alone) is better than integrating the historical error-based method that also makes the model give attention to early days predictions depending on their losses.

Fig. 4 shows an example of the RUL predictions to one HDD and the several methods applied. It is clear in the figures that the equal weighting method has issues predicting both the beginning and the end life of the HDD. However, employing the increasing and decreasing methods improves the early and last days, respectively. It is also clear that applying the historical error-based method improves the overall predictions, even when employed by itself. One interesting aspect of this example is that, mainly when the decreasing method is not applied, the model tends to be alarmist in the last days, predicting a lot of days as the failure point. This endorses the hypothesis that the close-to-failure time frame is hard to predict and shows that an intervention - such as sample weighting methods - can be useful.

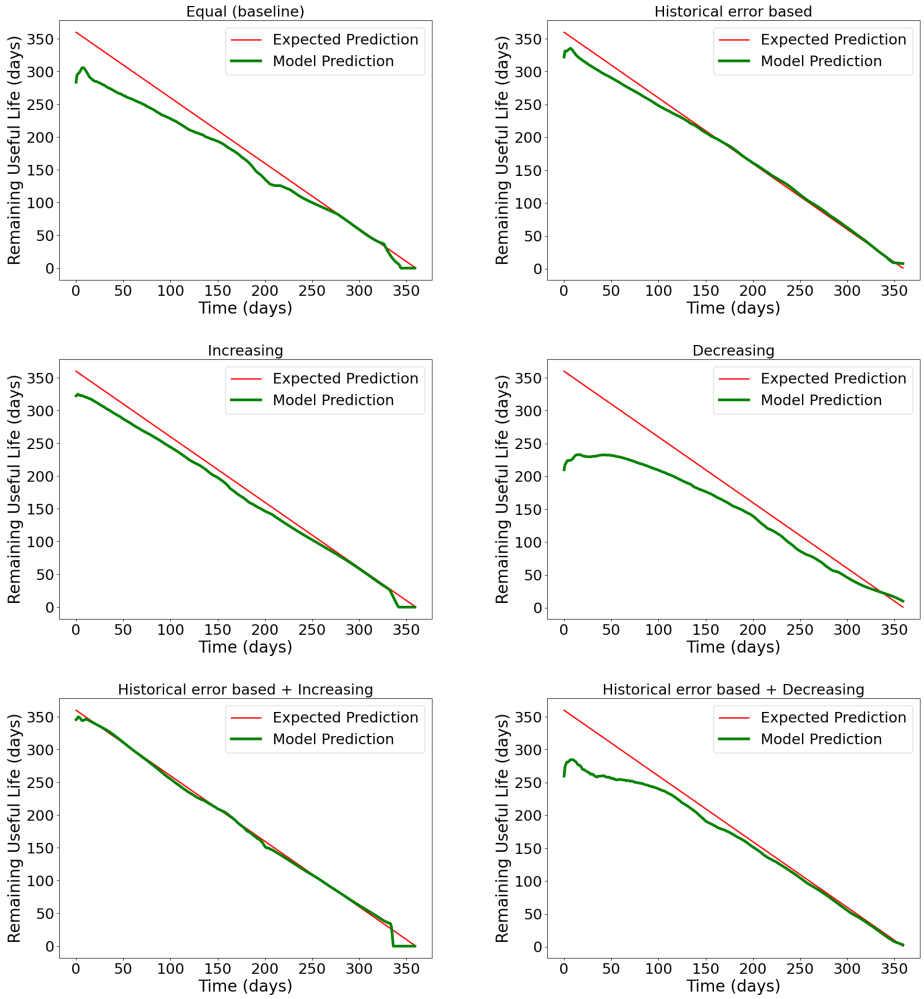

Fig. 4: Expected RUL and prediction results of the experimented models for the HDD S301GN45.

5. CONCLUSION

In this article, we proposed several methods of sample weighting to improve the accuracy of predictions of the HDD's remaining useful life, addressing the need to do better predictions near the end/beginning of its life. We conducted experiments that demonstrate the effectiveness of these methods for their respective purposes.

To obtain better predictions for the last 360 days of life of HDDs, in general, the historical error based combined with the increasing method proved to be the best method, getting an improvement of about 45% for MSE metric when compared to the baseline (Equal) method. Still, all methods seem to work in what they are designed for.

In future works, we are going to evaluate the impact of changing the hyperparameters of the methods and new combination sets of these methods on the accuracy.

ACKNOWLEDGMENT

This research was partially funded by Lenovo, as part of its R&D investment under Brazilian Informatics Law, by CAPES grants 88887.609134/2021-00 and 88887.609129/2021, and CNPQ grants 307323/2022-6 and 316729/2021-3.

REFERENCES

Backblaze. Hard drive data and stats. https://www.backblaze.com/b2/hard-drive-test-data.html, 2023. Accessed: 2023-02-13.

Cahyadi and Forshaw, M. Hard disk failure prediction on highly imbalanced data using lstm network. In *2021 IEEE International Conference on Big Data (Big Data)*. pp. 3985–3991, 2021.

Chaves, I. C., de Paula, M. R. P., Leite, L. G., Queiroz, L. P., Gomes, J. P. P., and Machado, J. C. Banhfap: A bayesian network based failure prediction approach for hard disk drives. In *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*. IEEE, pp. 427–432, 2016.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation* 9 (8): 1735–1780, 1997.

Hu, L., Han, L., Xu, Z., Jiang, T., and Qi, H. A disk failure prediction method based on lstm network due to its individual specificity. *Procedia Computer Science* vol. 176, pp. 791–799, 2020. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.

Lima, F. D. S., Pereira, F. L. F., Chaves, I. C., Gomes, J. P. P., and Machado, J. C. Evaluation of recurrent neural networks for hard disk drives failure prediction. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, pp. 85–90, 2018.

Lima, F. D. S., Pereira, F. L. F., Chaves, I. C., Machado, J. C., and Gomes, J. P. P. Predicting the health degree of hard disk drives with asymmetric and ordinal deep neural models. *IEEE Transactions on Computers* 70 (2): 188–198, 2021.

Murray, J. F., Hughes, G. F., and Kreutz-Delgado, K. Machine learning methods for predicting failures in hard drives: A multiple-instance application. *J. Mach. Learn. Res.* vol. 6, pp. 783–816, 2005.

Olah, C. Understanding lstm networks, 2015. [Online; accessed 2017-04-26].

Ottem, E. and Plummer, J. Playing it smart: The emergence of reliability prediction technology. Tech. rep., Technical report, Seagate Technology Paper, 1995.

Pereira, F. L. F., Bucar, R. C. B., Brito, F. T., Gomes, J. a. P. P., and Machado, J. C. Predicting failures in hdds with deep nn and irregularly-sampled data. In *Intelligent Systems: 11th Brazilian Conference, BRACIS 2022, Campinas, Brazil, November 28 – December 1, 2022, Proceedings, Part II*. Springer-Verlag, Berlin, Heidelberg, pp. 196–209, 2022.

Pinheiro, E., Weber, W.-D., and Barroso, L. A. Failure trends in a large disk drive population. In *5th USENIX Conference on File and Storage Technologies (FAST 07)*. USENIX Association, San Jose, CA, 2007.