# Using Autoencoders for Anomaly Detection in Hard Disk Drives

Francisco Lucas F. Pereira
*LSBD/DC/MDCC*
*Federal University of Ceará*
Fortaleza, Brazil
lucas.falcao@lsbd.ufc.br

Iago Castro Chaves
*LSBD/DC/MDCC*
*Federal University of Ceará*
Fortaleza, Brazil
iago.chaves@lsbd.ufc.br

João Paulo P. Gomes
*LSBD/DC/MDCC*
*Federal University of Ceará*
Fortaleza, Brazil
joao.pordeus@lsbd.ufc.br

Javam C. Machado
*LSBD/DC/MDCC*
*Federal University of Ceará*
Fortaleza, Brazil
javam.machado@lsbd.ufc.br

***Abstract*—Nowadays, predicting failures in Hard Disk Drives (HDD) is of key importance for storage service providers and end users. Being able to detect in advance that a disk is going to fail may enable maintenance actions that can avoid severe data losses. For that reason, many researchers had devoted attention to this research topic. Recently, several authors have reported promising results by using attributes collected by the SMART (*Self-Monitoring, Analysis and Reporting Technology*) system along with machine learning methods. Although the best results were obtained by supervised machine learning methods, it is important to notice that data from degraded HDDs is scarce. Hence, anomaly detection methods arise a promising solution. Among such methods, recent works reported that reconstruction based anomaly detection algorithms had the best performance on HDDs fault detection. In line with such results, in this paper we aim to further investigate the performance of such methods. We conducted tests with classical PCA based methods and neural autoencoder based methods. In addition to testing with the popular reconstruction based autoencoder method we also evaluated a method that analyzes the distribution of the latent space. Additionally we propose a simple formulation to combine both methods. On the basis of our experiments, we verified that autoencoder based methods had the best performances according to the two evaluation metrics. Among such methods, the combination approach had the best overall performance.**

***Index Terms*—autoencoder, anomaly detection, hard disk drives, fault detection**

## I. Introduction

Nowadays, Hard Disk Drives (HDD) are the prevalent storage technology in big data environments [1]. Although solid state devices are lighter, faster and less prone to degradation, its high cost and difficult recovery process makes HDDs still a valid alternative specially for storage service providers.

It is well known that, given that HDDs comprise a set of moving parts, a degradation of such components is expected and that can lead to severe data losses. Therefore, being capable of identifying degraded HDDs turns to be highly desirable. Aiming to pursue such objective, HDDs manufacturers developed *Self-Monitoring, Analysis and Reporting Technology* (SMART). SMART is a system that monitors several disk parameters and check if any of them exceed pre-defined thresholds. Although SMART is widely used, its failure detection rate is low, typically ranging from of 3% to 10% [2].

In recent years, many researchers have devoted efforts towards providing a more reliable failure prediction method for HDDs. In most of such works, the SMART attributes are combined with machine learning methods. One of the first works was presented by Murray *et al.* in [2]. In this paper, the authors tested several machine learning methods and verified that Support Vector Machine (SVM) had the best fault detection rate. Chang Xu *et al.* [3] formulated the problem as a classification between several health states and not only faulty and healthy states as presented in [2]. Chang Xu and its co-authors used the SMART attributes as inputs for a Recurrent Neural Network (RNN). The same problem was addressed by Chaves *et al.* [4] by using a Bayesian Network. Also, Lima *et al.* [5] had promising results with deep neural models.

It is worth noting that in previous works, the authors used data from degraded and healthy HDDs. However, in many real situations, only data from healthy disk is available [6]. For that problem, several authors proposed methods based on anomaly detection algorithms. Hughes *et al.* [7] reported a detection rate of 32% (with a 0.2% false alarm rate) by using a non-parametric hypothesis test to detect anomalies. Queiroz *et al.* [8], [9] and Wang *et al.* [10], [11] used anomaly detection methods based on density estimation algorithms and were able to outperform the results of Hughes *et al.* [7]. For the same task, a deeper investigation on the performance of various types of anomaly detection methods is presented in [6]. The authors evaluated the performances of 9 methods that were divided in 3 different approaches: density methods, boundary methods and reconstruction methods. The conclusion of this study was that reconstruction methods had the best compromise between fault detection and false alarm rates. The authors found that both PCA and neural autoencoders had very similar performances.

Motivated by such results, in this work we aim to expand the investigation of reconstruction based anomaly detection methods for fault detection in HDDs in the following directions:

- We evaluated not only the reconstruction error as a fault indicator but also the distribution of the data in the latent space. Such approach is widely used in PCA for industrial applications [12] but indeed less popular when designing fault detection methods based on autoencoders.
- We proposed a simple way to combine fault indicators

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

provided by the reconstruction error and the probability distribution of the latent space in autoencoders. The proposed approach is similar to the combination method for PCA proposed in [13].

The remainder of this paper is divided as follows. Section 2 presents a brief description of the most relevant related papers. Section 3 shows the basic concepts that are necessary to understand our proposal. Such proposal is presented in Section 4 along with the other methods used for a comparative analysis. The experiments and the associated analysis are shown in Section 5 and the final remarks are presented in Section 6.

## II. Related Work

Fault detection is an active research topic that has stimulated the development of various methods and its application in many industrial environments [14]. One of the most common approaches is based on finding a low dimensional projection of the original data that captures the major relations among the process variables [15]. Most well-known methods are principal component analysis (PCA), independent component analysis (ICA), partial least squares (PLS) and autoenconders.

After estimating the reduced subspace (latent space), two anomaly detection procedures are widely used. The first procedure consists of estimating a transformation that can project the data from the latent space to the original space. Then, the so called reconstruction error can be computed. Thus, an anomaly (fault) is detected if the reconstruction error is higher than a given threshold. One can find several recent works that follow this approach and use both linear [15]–[17] and nonlinear [14], [18], [19] dimensionality reduction methods.

In the second procedure, a similarity/dissimilarity measure is computed between a test sample and the training data. Such measure is computed in the latent space. The Euclidean distance is often used as a metric. A fault is detected if the similarity/dissimilarity measure violates a threshold. It is important to mention that the two procedures can be combined as presented in [13]. In this work, the authors propose a method to combine the two approaches for PCA. Although the results obtained in this paper are promising, methods for combining the two procedures are indeed rare and often limited to linear dimensionality reduction methods.

## III. Theoretical Background

### A. Principal Component Analysis (PCA)

Principal component analysis is a dimensionality reduction technique that combines a set of correlated variables to create a new set of uncorrelated variables while maintaining most of the variance of the original variables.

Let $x \in \mathcal{R}^m$ denote a sample vector of $m$ variables. Assuming that there are $n$ samples, a matrix $X \in \mathcal{R}^{n\times m}$ comprises the set of samples where each samples is in a row. Consider that the data have zero mean and unit variance. PCA determines the transformation of $X$ that captures most of the variance in $X$ and concentrates it in several dimensions of the transformed data. For that, the matrix $X$ can be decomposed as follows:

$$X = TP^T \tag{1}$$

where $T = [t_1, t_2, ..., t_m] \in \mathcal{R}^{n\times m}$ are the principal components and $P = [p_1, p_2, ..., p_m] \in \mathcal{R}^{m\times m}$ are the loading vectors. The loading vectors are the eigenvectors associated with the eigenvalues of the covariance matrix of $X$. The eigenvectors represent the vectors that transform the original data to each of the principal components and the associated eigenvalues represent the amount of variance represented in each principal component.

*1) Fault Detection Using PCA:* Fault detection methods using PCA are based on Hotelling's $T^2$ and $Q$ statistics. In $Q$ statistics, the objective is to measure of the difference (residual) between a sample $x_i$ and its projection onto the $l$ principal eigenvectors retained in the model. The residual for sample is given by:

$$r_i = x_i(I - P_l {P_l}^T) \tag{2}$$

and the magnitude of the residual is given by:

$$Q = r_i {r_i}^T \tag{3}$$

The confidence limits for $Q$ can be obtained from its approximated distribution [20].

Hotelling's $T^2$ statistic provides an indication of unusual variability within the reduced space (latent space). It represents the length of the projection of a given sample into the space spanned by the $l$ principal components. More formally, $T^2$ statistics is given by:

$$T^2 = {x_i}^T P_l \lambda^{-1} {P_l}^T x_i \tag{4}$$

where $P_l$ is a matrix of the loading vectors comprising the first $l$ principal eigenvectors and $\lambda$ is a diagonal matrix containing the first $l$ eigenvalues. The confidence limits for $T^2$ can be obtained analytically since it follows a F-distribution [20].

### B. Autoencoders

The autoencoders are unsupervised learn techniques for compact the input into lower dimension latent space. Autoencoders can be divided into two parts: encoder and decoder.

The encoder tries to learn a function $h_{W,b}(x) = g(f(x)) \approx x$ that compress the input into a latent space. In other words, the function tries to map the input to a similar representation with lower dimension. The decoder aims to reconstruct the input from the latent space representation. Between the encoder and decoder we have a bottleneck that represents the data in a different dimension. The bottleneck dimension is a constraint that defines the dimension of the lower dimension representation.

In the encoder process a function $f$ is applied to the input $x \in \mathbb{R}^m$ obtaining the compressed latent vector. After that the decoder applies the $g$ function to decode $f(x)$ to $x' \in \mathbb{R}^n$.

The entire network can be constructed by minimizing the reconstruction error $\mathcal{L}(x, x')$ + regularizer. In the literature commonly the $L1$ regularization, $L2$, or KL-Divergence are

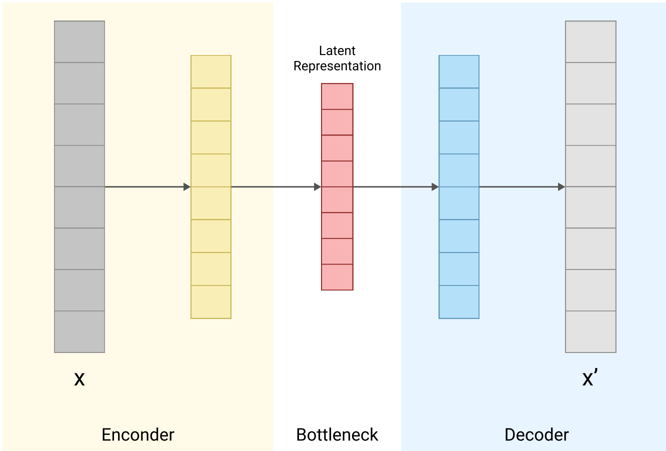


Fig. 1: Illustration of an Autoencoder.

used as regularizers, so they act to avoid the memorization or overfitting.

## IV. Methodology

In this section we describe the our fault detection methodology based on autoencoders. We used both the reconstruction error and the distribution in the latent space. After that we briefly describe the dataset and the experimental procedure.

### A. Fault detection based on autoencoders

To detect faults in HDDs we decided use autoencoder neural networks in two different ways. In the first approach we compute the reconstruction error of a given test sample. A fault is indicated if the reconstruction error exceeds a given threshold. This approach is common and has been used in other applications in works like [21], [22] and [23]. Throughout this paper we will refer this approach as Reconstruction Error method (REA). A brief description of REA is shown in Algorithm 1.

In the second method we aim to detect faults by checking if a new sample is different from the training samples in the latent space. The idea is similar to the $T^2$ statistics in PCA. For that, after training the autoencoder, we compute the mean and covariance matrix of the training samples in the latent space. With this information, we can calculate the Mahalanobis distance of a test sample after a transformation to the latest space. A fault is detected when the distance exceeds a threshold. This approach will be named Distance in the Latent Space method (DLS) and is shown in Algorithm 2.

In this work, we also propose a method that combine both metrics in a balanced way. According to Yue and Qin [13], $T^2$ and $Q$ exhibit complementary behaviors and its combination is beneficial. They proposed the following combined index.

$$\varphi = \frac{Q}{\delta^2} + \frac{T^2}{\chi^2} \tag{5}$$

where $\delta^2$ and $\chi^2$ are the fault detection thresholds for $Q$ and $T^2$ respectively. Roughly speaking, the main idea is to balance both metrics with such normalizing coefficients. These normalizing coefficients (thresholds) are calculated so that the same ratio of points are below each threshold. Using a similar idea, we can obtain the normalizing coefficients for our metrics by constructing an empirical CDF with the training samples. The resulting formulation is shown in Eq 6 and the algorithm is shown in Algorithm 3.

$$\vartheta = \frac{REA}{REA_T} + \frac{DLS}{DLS_T} \tag{6}$$

where $REAT_T$ and $DLS_T$ are anomaly thresholds obtained using the empirical CDF for a given probability. $\vartheta$ is the resulting anomaly detection index.

**Algorithm 1** Dissimilarity Measure using the Reconstruction Error approach.

$D_{healthy}, D_{faulty} \leftarrow get\_class\_data(D)$
$D_{train}, D_{healthy\_test} \leftarrow split(D_{health})$
$D_{test} \leftarrow D_{faulty} + D_{healthy\_test}$
$AE \leftarrow train\_autoencoder(D_{train})$

$distances\_list \leftarrow empty\_list()$
**for** $x \in D_{test}$ **do**
    $x' \leftarrow AE(x)$
    $REA \leftarrow x - x'$
    $append\_to\_list(distances\_list, REA)$
**end for**
**return** $distances\_list$

**Algorithm 2** Dissimilarity Measure using the Mahalanobis Distance in the Latent Space approach.

$D_{healthy}, D_{faulty} \leftarrow get\_class\_data(D)$
$D_{train}, D_{healthy\_test} \leftarrow split(D_{health})$
$D_{test} \leftarrow D_{faulty} + D_{healthy\_test}$
$AE \leftarrow train\_autoencoder(D_{train})$

$\ell_{train} \leftarrow D_{train}$ in the latent space of $AE$
$means, cov \leftarrow get\_means(\ell_{train}), get\_cov(\ell_{train})$

$distances\_list \leftarrow empty\_list()$
**for** $x \in D_{test}$ **do**
    $lx' \leftarrow x$ in the latent space of $AE$
    $DLS \leftarrow Mahalanobis(lx', mean, cov)$
    $append\_to\_list(distances\_list, DLS)$
**end for**
**return** $distances\_list$

### B. Dataset

In order to evaluate the performance of the methods, a set of experiments were conducted using a dataset publicly provided by the Backblaze Company [24]. The dataset contains daily SMART observations of thousands of Hard Disk Drives from a large number of manufacturers ranging from April 2013 to December 2016. These SMART observations are retrieved until the disk stops working or until it has showed some

**Algorithm 3** Dissimilarity Measures using the combination of REA and DLS approach.

$D_{healthy}, D_{faulty} \leftarrow get\_class\_data(D)$
$D_{train}, D_{healthy\_test} \leftarrow split(D_{health})$
$D_{test} \leftarrow D_{faulty} + D_{healthy\_test}$
$AE \leftarrow train\_autoencoder(D_{train})$

$\ell_{train} \leftarrow D_{train}$ in the latent space of $AE$
$means, cov \leftarrow get\_means(\ell_{train}), get\_cov(\ell_{train})$
$latent\_threshold \leftarrow$ calculate the empirical threshold for the Mahalanobis distances of $\ell_{train}$.

$err_{train} \leftarrow$ calculate the reconstruction error for all training data
$rec\_threshold \leftarrow$ calculate the empirical threshold for the reconstruction errors of $err_{train}$.

$distances\_list \leftarrow empty\_list()$
**for** $x \in D_{test}$ **do**
  $x' \leftarrow AE(x)$
  $REA \leftarrow x - x'$

  $lx' \leftarrow x$ in the latent space of $AE$
  $DLS \leftarrow Mahalanobis(lx', mean, cov)$

  $combin\_measure \leftarrow \frac{REA}{rec\_threshold} + \frac{DLS}{latent\_threshold}$

  $append\_to\_list(distances\_list, combin\_measure)$
**end for**
**return** $metrics$

indication that it will stop to work soon, and the disk is marked as failed in the dataset.

It is assumed that Hard Disk Drives from the same manufacturer model have similar degradation over time. Therefore, for the experiments, the two HDD models with more data were selected: the Seagate ST4000DM000 and Seagate ST3000DM001.

From ST4000DM000, there are 36,555 disks, of which 1,729 have failed. From this dataset, 32 were excluded because their observations were interrupted without a label indicating a failure or because they had more observations submitted after being labeled with failure. Also, it was selected disks that lived at least 360 days, resulting in 907 instances used.

From model ST3000DM001, we had a set of 4,707 disks and 1,357 failed. Because of the same inconsistencies described to the other model, 345 disks were excluded. Also, due to the data being less plentiful than the other manufacturer model, we selected a subset of the remaining failed disks that had at least 90 days of continuous monitoring. After such procedure, we ended up with 786 disks.

For these datasets, a day is considered faulty if it is in the last 30 days of the HDD life, and healthy if is is in any time before the last 30 days, as illustrated in Figure 2. For each of these disks, a day from the healthy span was selected at

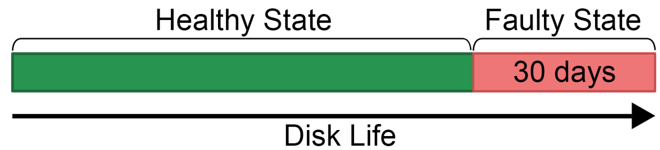


Fig. 2: Hard Disk Drives life stages until the day of its failure. The last 30 days are considered faulty.

random to be a healthy sample and a day from the last 30 days was selected at random to be a faulty sample. Therefore, the goal is to classify if a HDD is in its last month of life. The healthy dataset was split in 70% for training and 30% for testing. Also, since we are not training with the faulty data, all faulty data is used for testing.

The selected SMART attributes were the raw values of the SMARTs available to the selected models ST4000DM000 and ST3000DM001, presented in Table I. An explanation of the SMART attributes can be found in [25].

| S.M.A.R.T ID | Attribute Name |
|---|---|
| 1 | Read Error Rate |
| 3 | Spin-Up Time |
| 4 | Start/Stop Count |
| 5 | Reallocated Sectors Count |
| 7 | Seek Error Rate |
| 9 | Power-On Hours |
| 10 | Spin Retry Count |
| 12 | Power Cycle Count |
| 183 | SATA Downshift Error Count |
| 184 | End-to-End error / IOEDC |
| 187 | Reported Uncorrectable Errors |
| 188 | Command Timeout |
| 189 | High Fly Writes |
| 190 | Temperature Difference |
| 191 | G-sense Error Rate |
| 192 | Unsafe Shutdown Count |
| 193 | Load Cycle Count |
| 194 | Temperature |
| 197 | Current Pending Sector Count |
| 198 | Uncorrectable Sector Count |
| 199 | UltraDMA CRC Error Count |
| 240 | Head Flying Hours |
| 241 | Total LBAs Written |
| 242 | Total LBAs Read |

TABLE I: SMART attributes used in the experiments.

### C. Experimental setup

In all experiments the fault detection methods were trained with healthy HDD data (training set) and tested with both healthy and degraded data (test set). To evaluate the performances we computed two metrics: the Area Under the ROC Curve (AUC) and the True Positive Rate (TPR) for a False Positive Rate (FPR) of 10%.

The AUC metric is a usual choice for comparing anomaly detection methods since it provides a way to compare such methods without depending on the choice of a threshold for each method. The second metric is crucial for the application since our objective is to design a monitoring system that can detect faults with a reduced number of false alarms (FPR).

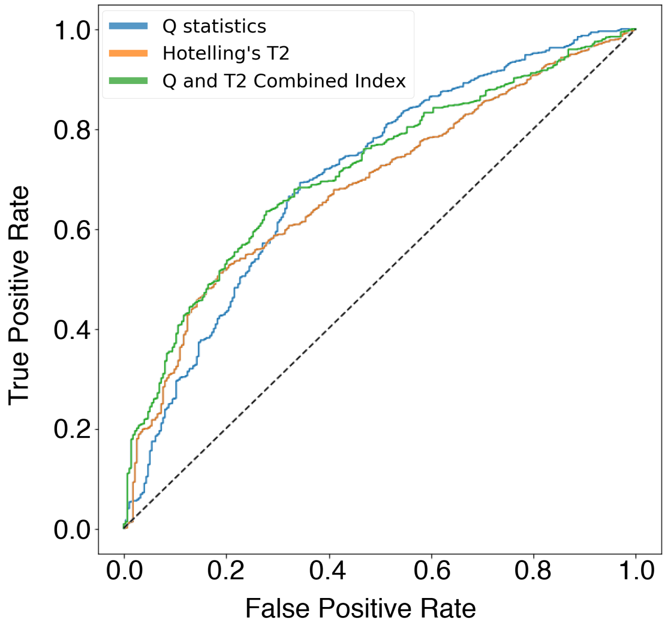


Fig. 3: PCA ROC Curves for the HDD model ST4000DM000.

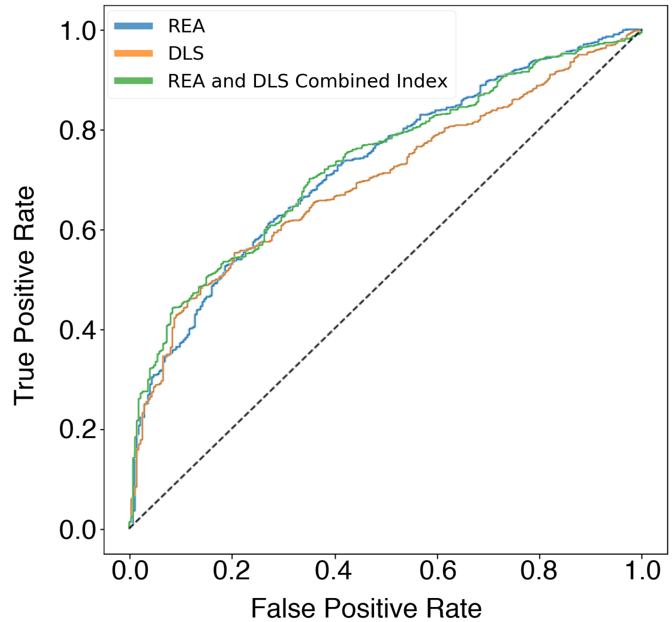


Fig. 4: Autoencoder ROC Curves for the HDD model ST4000DM000.

False alarms may result in unplanned maintenance actions thus increasing the costs of storage service providers. We repeated all tests 20 times and computed the average values of both metrics. All experiments were executed using scikit-learn package [26] version 0.17 and Tensorflow 1.7.0 [27] .

For the PCA method, we performed the transformation and selected the eigenvectors that resulted in features that preserve 90% of the variance, resulting in 8 features out of the 24 described in Table I. For the Autoencoders, it was trained a neural network architecture with hidden layers of size (15-8-15) and a output layer of size 24 (the number of dimensions of the input), with the ReLU activation function and the backpropagation algorithm with L2 regularization.

## V. Experiments and Results

To further investigate the feasibility of reconstruction based anomaly detection methods in the task of HDDs fault detection we decided to evaluate the performance of classical PCA based methods, $Q$ and $T^2$, and the combination of both metrics proposed in [13]. The use of such baselines is fundamental since the authors in [6] reported good results for $Q$ and the literature shows that the combination of $Q$ and $T^2$ often outperforms each metric individually. The REA method was evaluated in [6] and was evaluated once again in this work. Along with that, we tested DLS and the proposed combination of both metrics.

The ROC curves for all methods computed for ST4000DM000 and ST3000DM001 are shown in Figs 3, 4, 5 and 6. The average values of all numerical metrics computed on 20 repetitions are shown in Table II and III.

By observing all ROC curves and the average metrics values, one can see that the combination methods (PCA based and Autoencoder based) had the best performances when

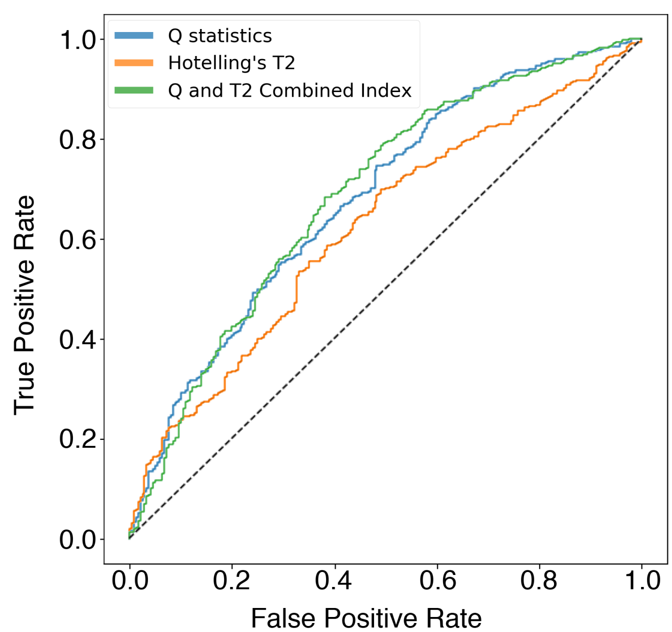


Fig. 5: PCA ROC Curves for the HDD model ST3000DM001.

| Method | Dissimilarity Measure | Average AUC | Average TPR at 10% FPR |
| --- | --- | --- | --- |
| PCA | Q statistics | 0.7053 | 0.2619 |
| | Hotelling's T2 | 0.6873 | 0.3815 |
| | Q and T2 Combined Index | 0.7143 | 0.3646 |
| Autoencoder | Reconstruction Error | 0.7217 | 0.4013 |
| | Distance in the Latent Space | 0.6962 | 0.3645 |
| | Combination of DLS and REA | **0.7303** | **0.4206** |

TABLE II: Results for disk ST4000DM000.

| Method | Dissimilarity Measure | Average AUC | Average TPR at 10% FPR |
| --- | --- | --- | --- |
| PCA | Q statistics | 0.6805 | 0.2402 |
| | Hotelling's T2 | 0.6218 | 0.2145 |
| | Q and T2 Combined Index | 0.6861 | 0.2676 |
| Autoencoder | Reconstruction Error | 0.6678 | 0.2552 |
| | Distance in the Latent Space | 0.6527 | 0.2494 |
| | Combination of DLS and REA | **0.6994** | **0.2809** |

TABLE III: Results for disk ST3000DM001.

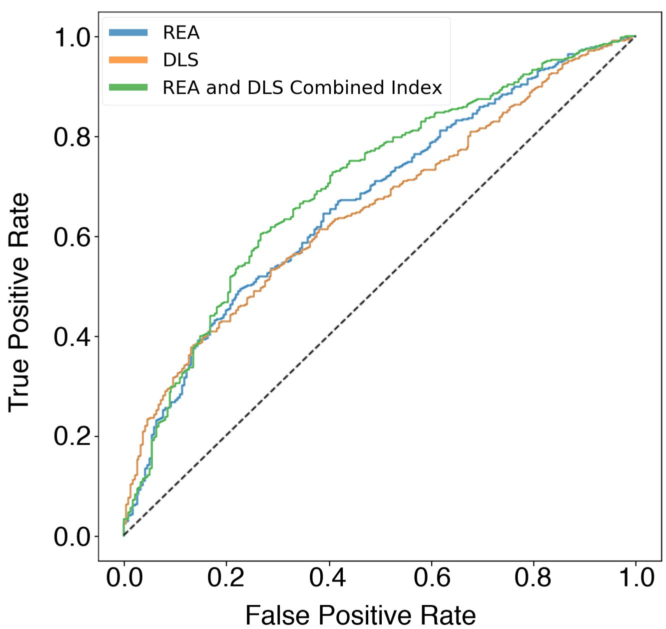


Fig. 6: Autoencoder ROC Curves for the HDD model ST3000DM001.

analyzing both AUC and TPR at $10\%$ FPR metrics. The good performance of the combination methods can be noticed in all ROC curves since the curves of combination methods exhibit the highest (or near) TPR values for all FPR. Among the combination methods, the autoencoder had the best overall results.

It is interesting to notice that all autoencoder based methods outperformed its PCA based counterparts. That fact may indicate that the problem of modeling the normal behavior of HDDs is nonlinear.

## VI. Conclusion

In this paper we present an evaluation of autoencoder based anomaly detection methods for fault detection in hard disk drives. We conducted experiments with the well-known reconstruction based approach and also adapted the idea of analyzing the latent space of dimensionality reduction methods. Such approach is inspired by the $Q$ statistics used with PCA. We also proposed a simple way to combine both approaches to generate a single anomaly detection index.

All experiments were performed on data from more than $1500$ HDDs of two models. The autoenconder methods were compared to classical PCA based algorithms and showed better results. Such fact may indicate that the problem of modeling the behavior of healthy HDDs is nonlinear. Among all autoencoder based approaches, the combination method showed the best performance.

In future work, we intend to evaluate the performance of variational autoencoders (VAE) for the same task. We hipothetize that VAE might show good results since that tend to force the distribution of the latent space to be normal, thus avoiding the occurrence probability mass in unknown regions of the space.

## Acknowledgment

This work was partially funded by Lenovo, as part of its R&D investment under Brazil's Informatics Law. The authors also acknowledge the support of CNPq (grant 131486/2018-7) and LSBD/UFC.

## References

[1] Z.-S. Ye, M. Xie, and L.-C. Tang, "Reliability evaluation of hard disk drive failures based on counting processes," *Reliability Engineering System Safety*, vol. 109, pp. 110 – 118, 2013.
[2] 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, pp. 783–816, 2005.

[3] 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 Transactions on Computers*, vol. 65, no. 11, pp. 3502–3508, 2016.

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

[5] F. D. dos Santos Lima, F. L. Falcão Pereira, I. Castro Chaves, J. P. Pordeus Gomes, and J. de Castro Machado, "Evaluation of recurrent neural networks for hard disk drives failure prediction," in *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, Oct 2018, pp. 85–90.

[6] F. Pereira, D. Teixeira, J. P. Gomes, and J. Machado, "Evaluating one-class classifiers for fault detection in hard disk drives," in *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, Oct 2019, pp. 586–591.

[7] G. Hughes, J. Murray, K. Kreutz-Delgado, and C. Elkan, "Improved disk-drive failure warnings," *Reliability, IEEE Transactions on*, vol. 51, no. 3, pp. 350–357, 2002.

[8] L. P. Queiroz, F. C. M. Rodrigues, J. P. P. Gomes, F. T. Brito, I. C. Chaves, M. R. P. Paula, M. R. Salvador, and J. C. Machado, "A fault detection method for hard disk drives based on mixture of gaussians and non-parametric statistics," *IEEE Transactions on Industrial Informatics*, 2016.

[9] L. P. Queiroz, J. P. P. Gomes, F. C. M. Rodrigues, F. T. Brito, I. C. Chaves, L. G. M. Leite, and J. C. Machado, "Fault detection in hard disk drives based on a semi parametric model and statistical estimators," *New Generation Computing*, vol. 36, no. 1, pp. 5–19, Jan 2018.

[10] Y. Wang, Q. Miao, E. W. Ma, K.-L. Tsui, and M. G. Pecht, "Online anomaly detection for hard disk drives based on mahalanobis distance," *IEEE Transactions on Reliability*, vol. 62, no. 1, pp. 136–145, 2013.

[11] Y. Wang, E. Ma, T. Chow, and K.-L. Tsui, "A two-step parametric method for failure prediction in hard disk drives," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 419–430, 2014.

[12] L. Chiang, E. Russell, and R. Braatz, *Fault Detection and Diagnosis in Industrial Systems*, 01 2001.

[13] H. Yue and J. Qin, "Reconstruction-based fault identification using a combined index," *Industrial and Engineering Chemistry Research*, vol. 40, pp. 4403–4414, 10 2001.

[14] P. Park, P. Di Marco, H. Shin, and J. Bang, "Fault detection and diagnosis using combined autoencoder and long short-term memory network," *Sensors*, vol. 19, p. 4612, 10 2019.

[15] B. Mnassri, E. M. E. Adel, B. Ananou, and M. Ouladsine, "Fault detection and diagnosis based on pca and a new contribution plot," *IFAC Proceedings Volumes*, vol. 42, no. 8, pp. 834 – 839, 2009, 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.

[16] B. P. Leao, J. P. P. Gomes, R. K. H. Galvão, and T. Yoneyama, "Aircraft flap and slat systems health monitoring using statistical process control techniques," *2009 IEEE Aerospace conference*, pp. 1–8, 2009.

[17] R. Qi and J. Zhang, "Process fault detection and reconstruction by principal component analysis," in *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Aug 2019, pp. 594–599.

[18] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Processing*, vol. 130, 07 2016.

[19] M. Kallas, G. Mourot, D. Maquin, and J. Ragot, "Diagnosis of nonlinear systems using kernel principal component analysis," *Journal of Physics: Conference Series*, vol. 570, no. 7, p. 072004, dec 2014.

[20] T. Villegas, M. J. Fuente, and M. Rodríguez, "Principal component analysis for fault detection and diagnosis. experience with a pilot plant," in *Proceedings of the 9th WSEAS International Conference on Computational Intelligence, Man-machine Systems and Cybernetics*, ser. CIMMACS '10. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2010, pp. 147–152.

[21] A. Anaissi and S. M. Zandavi, "Multi-objective autoencoder for fault detection and diagnosis in higher-order data," *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2019.

[22] W. Zhang and X. Tan, "Combining outlier detection and reconstruction error minimization for label noise reduction," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Feb 2019, pp. 1–4.

[23] S. Chang, B. Du, and L. Zhang, "A sparse autoencoder based hyperspectral anomaly detection algorihtm using residual of reconstruction error," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, July 2019, pp. 5488–5491.

[24] Backblaze. (2016, apr) Hard drive data and stats. [Online; accessed 2019-05-24]. [Online]. Available: https://www.backblaze.com/b2/hard-drive-test-data.html

[25] Kingston, "Smart attribute details," Kingston Technology Corporation, Tech. Rep., 2015.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.