2017 Brazilian Conference on Intelligent Systems

Transfer Learning for Bayesian Networks with Application on Hard Disk Drives Failure Prediction

Francisco Lucas F. Pereira, Fernando Dione S. Lima, Lucas G. M. Leite, Joao Paulo P. Gomes, Javam C. Machado
LSBD - Department of Science Computer
Federal University of Ceará, Fortaleza, Brazil
Email: {lucas.falcao, fernando.lima, lucas.goncalves, joao.pordeus, javam.machado}@lsbd.ufc.br

***Abstract*—Predicting failure in Hard Disk Drives (HDD) is of fundamental importance for data loss avoidance as well as to lower the downtime costs of a system. As a consequence, an increasing effort may be observed from both universities and industry to find suitable failure prediction methods. Despite the encouraging performances achieved by various methods one important aspect that shall be noticed is the lack of data available to build reliable models. Considering the HDD failure prediction task, this problem arises with the numerous HDD models. To overcome such problem, transfer learning strategies offer a valid alternative since it can be used to transfer learning from HDD model with enough data to build failure prediction methods for HDD models with lack of data. In this work we evaluate several transfer learning strategies in the task of HDD failure prediction. Additionally we propose a new strategy to build information sources based on the clustering of similar HDD models. This approach may be a valid alternative when no HDD model has enough data to generate a reliable model. Results showed that all transfer learning scenarios can improve the performance of HDDs failure prediction methods, mainly for HDDs with very limited data. Moreover, the clustering-based information source also results in performance gains in all transfer methods and HDD models tested.**

I. INTRODUCTION

Being able to predict failures in Hard Disk Drives (HDD) has been one of the most pursued goals of HDD manufacturers in recent years. Currently, most HDDs are equipped with a monitoring system named *Self-Monitoring, Analysis and Reporting Technology* (SMART). The SMART consists in a computational system that monitors and collects data from the device and detect anomalies, commonly through a simple threshold based approach, which results in low Failure Detection Rates (FDR), ranging from 3% to 10% [1].

According to Vachtsevanos *et al.* [2], the term failure prognostics is often used to characterize two different tasks: incipient failure detection and Remaining Useful Life (RUL) estimation. In the former, a method is designed to detect an anomalous behavior of the system under study. This anomaly may indicate that an incipient failure occurred and although the system is still working, it will fail in a near future. Incipient failure detection methods for HDDs are presented in [3], [4], [5] and [6]. The latter consists of estimating the remaining time until the system stops working, the RUL. Usually this step is started after an anomaly is detected. Previous works on RUL estimation in HDDs are less frequent but can be found in [7] and [8].

An important aspect that shall be noticed when designing HDD failure prediction methods is the great variety of HDD models. It is expected that HDD from different manufacturers with different features (capacity, speed, size ...) may have several damage progression patterns. Hence, it may be necessary to build different prediction methods or adapt a baseline model to work for other HDDs. In this scenario, Transfer Learning (TL) methods are a possible alternative.

Transfer learning algorithms aim to alleviate data requirements of a task by leveraging data from related tasks [9]. Such methods have been successfully applied in various domains like natural language processing [10], image processing [11] and wireless communications [12]. In this paper, we evaluate various transfer learning strategies for the task of HDD failure prediction. HDD models with large amounts of data are used as sources and the generated knowledge is used to improve the performance of target models, from HDDs with limited data available. We used the Bayesian Network (BN) failure prediction method proposed in [7] as the baseline method. Additionally we propose the use of data from similar HDD's clusters as information source for transfer learning. This approach may be a valid alternative when no HDD model has enough data to generate a reliable model.

Results showed that all transfer learning approaches can improve the performance of HDDs failure prediction methods, mainly for HDDs with very limited data. Moreover, the clustering-based information source proposed also results in performance gains in all transfer methods and HDD models tested.

II. RELATED WORK

The task of HDD failure prediction has been addressed in many recent works. In [13], the authors designed a binary classification problem, where a given HDD was classified as normal or faulty using a Support Vector Machine (SVM). Wang *et al.* [5] achieved better results by modelling the problem as an anomaly detection task, where a statistical model is built using only fault-free HDDs. A faulty HDD is detected when its SMART features are significantly different from this statistical model. Further improvements were obtained in [3] and [4] where the authors used semi-parametric methods to build the statistical model of healthy HDDs and nonparametric

978-1-5386-2407-4/17 $31.00 © 2017 IEEE
DOI 10.1109/BRACIS.2017.64

statistics to quantify the difference between a given HDD and the statistical model.

Considering the task of RUL estimation, two recent papers were proposed in this topic. In Xu *et al.* [8], the authors used a Recurrent Neural Network (RNN) to classify HDDs in six health states according to their RUL. The RNN model inferred the health state of each HDD given its SMART features. The classification was conducted in datasets comprising the final working month of each HDD. Similarly, in [7] the authors also classified HDDs in health states according to their SMART attributes. However, the predictions were performed for longer time horizons. A BN based model was designed to infer HDD's health states up to two years before the failure.

As can be noticed, no previous work addressed the problem of reduced number of examples to build a RUL estimation model. In our study, we used the model proposed in [7] as a baseline RUL estimation method and tested several transfer learning strategies to handle this problem. Additionally, we proposed a novel way to build a source dataset for transfer learning in HDDs.

III. Theoretical Background

A. Bayesian Networks

A Bayesian Network (BN) [14] is a probabilistic graphical model that relates a set of random variables according to their conditional dependencies. A BN models the possible relations of cause and consequence of a given phenomenon from series of observations. Formally, a BN is a pair $B = (G, \Theta)$ such that $G = (X, E)$ is a directed acyclic graph whose set of vertices $X = \{X_1, X_2, \dots, X_n\}$ represents the random variables and the set of edges E represents the conditional dependencies between the variables represented by X. The dependencies are defined by the set of probability functions Θ. This set contains the parameter $\theta_{x_i|\pi_i} = P(x_i|\pi_i)$ for each $x_i \in X_i$ conditioned by π_i, that is, the set of parameters for X_i. The following equation presents the joint distribution defined by a BN over the set of random variables:

$$P(X_1, \dots, X_n) = \prod_{i=1}^{n} P(x_i|\pi_i) \qquad (1)$$

In its basic formulation, the BN is built upon probabilities estimated over discrete variables. However, in various applications only continuous variables are available. In such situations it is possible to use any discretization method such as the variant of the Minimum Description Length Principle (MDLP), proposed in [15], that considers a discretization based on a minimization heuristic entropy. The method proposed in [7] uses the entropy of the labels of classes to select a separation threshold of attributes, minimizing the entropy. The algorithm is then recursively applied to both fragments resulting from the separation.

B. Transfer Learning for Bayesian Networks

Transfer Learning (TL) aims to construct a model for a target task from data referring to this task and other related tasks, called source tasks. In this way, we can build more precise models from the enrichment of the dataset or model of the target task. In the case of Bayesian Networks, several transfer learning methods have been proposed and addressed to main topics: structure and parameter learning [16], [9].

Structure Learning is concerned in finding the best connections for the BN graph. In such problem, the objective is to identify the relation between variables using either the data from the target task and source task. Popular methods and a more detailed literature review can be seen in [17] and [16]. The task of parameters transfer learning in BN consists of building the Conditional Probability Tables (CPTs) using target and source data. In most cases, the structure of the BN is fixed (known) and is the same for both target and source tasks.

In this paper we are going to consider only the task of parameter transfer learning. For such task, three of the most popular methods are described in [16]. The methods, referred as Linear Aggregation , Logarithmic Aggregation and Distance Based Linear Pool are presented below.

Linear Aggregation: the new target probability of variable X, $P(X)$, is a weighted sum from the target and auxiliary source tasks, expressed as follows:

$$P(X) = k \sum_{i=1}^{m} w_i P_i(X) \qquad (2)$$

where $P_i(X)$ represents the conditional probability of the i-th model involving X, w_i is the weight of the i-th probability and k is a normalization factor.

Logarithmic Aggregation: the new target probability of variable X, $P(X)$ is a weighted product from the target and auxiliary source tasks, expressed as follows:

$$P(X) = k \prod_{i=1}^{m} P_i(X)^{w_i} \qquad (3)$$

where $P_i(X)$ represents the conditional probability of the i-th model involving X, w_i is the weight of the i-th probability and k is a normalization factor.

Distance Based Linear Pool: the method considers the confidence of the probability estimates from the auxiliary source tasks and their similarity with the target estimates. The method gives higher weight to the probability estimates that have higher confidence and are more similar to the target. The confidence is based on the size of each dataset. The DBLP can be divided in the following steps, where $p = P(X)$ denotes each individual probability in the CPTs:

1. Obtain the average probabilities of all datasets according to a confidence factor:

$$\overline{p} = k\sum_{i=1}^{m}(f_i.p_i) \tag{4}$$

where f_i is the confidence factor for each probability estimate and k is a normalization factor. The confidence factor depends on the size of the dataset used to estimate the CPT and is defined as follows:

$$f_i = \begin{cases} 1 - \frac{log(c_f)}{c_f}, & \text{if } c_f \geq 3 \\ 1 - \frac{c_f.log(3)}{3}, & \text{if } c_f < 3 \end{cases} \tag{5}$$

where $c_f = \frac{N}{2.T}$ is proportional to the expected error that depends on T, the number of entries in the CPT, and N is the number of cases in the dataset.

2. Obtain the minimum (d_{min}) and maximum (d_{max}) distance between the probability of the target and the above average.

3. Estimate the new conditional probabilities of the target model as follows:

$$p'_{target} = (1 - c_i)p_{target} + c_i\overline{p} \tag{6}$$

where the aggregation coefficients c_i express how much to consider from the CPTs of the other models. If the CPTs of the target network are similar to the average of all the CPTs, then more weight is given to the average, otherwise, more weight is given to the target's CPTs. This is expressed as follows:

$$c_i = (d_i - d_{min})\left(\frac{c_{max} - c_{min}}{d_{max} - d_{min}}\right) + c_{min} \tag{7}$$

where c_{max} and c_{min} are parameters to indicate how close we want to consider the influence of the other CPTs.

C. K-means for Datasets with Missing Values

Data completeness is one of the major assumptions of the k-means clustering algorithm. However, in many real world problems some data may have missing values due to various reasons. To overcome such problem, several methods have been proposed and achieved remarkable results [18], [19], [20].

In [20], the authors had one of the most successful attempts to solve this problem. The proposed method, named k-POD is built under the assumptions that each observation of a partially observed data matrix X is a noisy instance of a known centroid cluster and that the cluster membership of each observation is also known. Based on these assumptions, the algorithm estimates (impute) the missing entries in X through the use of the corresponding entries from their respective centroid. After filling the missing entries, the resulting matrix built from X is a complete data matrix that is then applied to k-means clustering. Finally, if after performing the k-means clustering, the resulting clusters assignments or centroids change, the imputation step is performed again. This process is repeated until convergence is achieved (i.e. no changes in the assignments). Further details of k-POD and its experimental evaluation can be seen in the original paper.

IV. Methodology

In this section, we present our methodology for applying Transfer Learning on HDD Failure Prediction. As explained above in Section III, with a group of trained Models, we may improve the performance of a target predictor transferring information from other Models if they perform a similar task.

Given a Prediction Model that is the target of the transfer process, there are two issues to solve: how to transfer (which methods) and from where to transfer (which strategies to select the sources). To deal with the first issue, we employ the three Transfer Learning methods for parameter learning explained in Section III. To address the second one, we propose strategies based on the size of available dataset for each disk model to choose the sources. In addition to that, we also explore a new strategy to improve transfer learning performance through the clustering of devices according to their manufacturer specifications. This effectively results in the creation of a new data source. This approach will be referred as clustering-based information source.

A. Failure Prediction Model

In this work, the baseline Failure Prediction Model will be a Bayesian Network with the structure presented in Figure 1. This structure is part of the BaNHFaP method, proposed in [7]. The input of the Prediction Model is an observation of 8 SMARTs representing the state of a specific HDD and its information of Power On Hours (POH), and the output is the expected Remaining Useful Life (RUL) of the disk.

The SMARTs used are 187, 5, 184, 7, 240, 190, 188 and 197, that can be respectively defined as Reported Uncorrectable Errors, Reallocated Sectors Count, End-to-End Error, Seek Error Rate, Head Flying Hours, Airflow Temperature, Command Timeout and Current Pending Sector Count. The RUL is discretized by quarters of the years, so the model is able to predict in which quarter of the following years the HDD will fail with some degree of certainty.

We are considering the same structure for all source and target models. As stated in [7], this structure showed good results for the same task addressed in this work. The only difference is that the authors in [7] built a single BN using data from several HDD models. In our work we aim to build an individual BN for each HDD model.

B. Transfer Learning Strategies

Once we have a group of trained Prediction Models, which in our case are Bayesian Networks, we perform the aggregation of information from part of this group to some target. Because the structure of the BNs is fixed, we are only left with the possibility of transferring knowledge from the CPTs of this

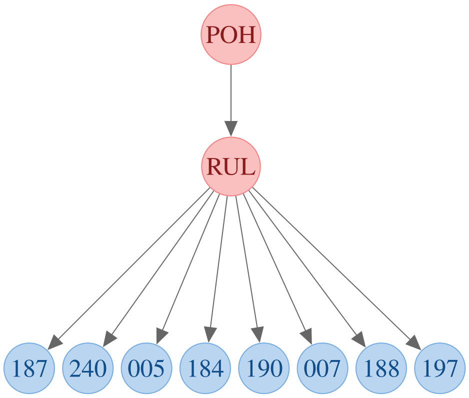

Fig. 1: Representation of the graph structure generated by the BaNHFaP method.

group of BNs to the target Model. The transfer of knowledge between CPTs is also known as Parameter Transfer.

Selecting the sources of the Parameter Transfer for a target disk can be a hard task. A naive approach is to choose all the Bayesian Networks that were trained with more data than the target one to be the sources. This simple strategy will be referred as **Strategy 1**.

In practice, some HDDs manufacturer models have a lot more data available than others, either because the disk model is older or because it is preferred by clients for being more cost-effective, for example. So another naive approach is to choose only the disk models whose available data is much larger than the other ones. For these disks, we expect them to have more accurate Failure Prediction Models. This strategy will be called **Strategy 2**.

C. A Clustering-based Information Source

Choosing all the disks with more available data than the target disk (Strategy 1) may not be the best option because, for example, the target disk can have very little data, so a lot of Prediction Models trained from disks with few data will also contribute to the final BN. Consequently, some BNs may disturb more than help in the transfer process, an effect known as Negative Transfer in the literature [21]. Transfer Learning methods such as DBLP try to ease this problem weighting the contribution of the sources according to their dataset sizes, but it can be bad if the given weights make its contributions very small or insignificant. This last problem also occurs when choosing only the disks with more available data (Strategy 2), because a lot of helpful information from the other disk models will not be considered.

Trying to prevent these problems, we propose a new way to consider the influence of the disks with few data, clustering them and concatenating the data of the resulting clusters to create new disk models.

We propose to use the manufacturer specification of the disks as features to execute the clustering process. We expect that HDDs with similar physical characteristics (specifications) have a similar degradation over time. The chosen specification features were capacity (in bytes), Revolution Per Minute (RPM), Data Transfer Interface, cache size, number of heads and internal disks, average latency for read/write operations and the device's form-factor.

As some disks do not have all those features available, we must employ a clustering algorithm that supports data entries with missing values. In our case, we apply the k-POD method [20] to do so. With the clusters produced by k-POD at hand, we can create data sources from the union of the data present in one or more of these clusters

We expect that the Failure Prediction Model trained with the data from the clustered disks to perform better than the individual Models, trained for each HDD model that belongs to the cluster. This way, the contribution of the Transfer Learning will be more efficient. Moreover, methods that considers the dataset size, like DBLP, will give more confidence to the clustered disks than if we would use the members of the cluster separately.

The **Strategy 3** uses the disks with a lot of data, like in Strategy 2, and also the clustered disks to be our sources in the Transfer Learning process.

V. Experimental Results

Our experiments were implemented in Python, using the Pandas 0.19.2 [22] and scikit-learn 0.18.1 [23] packages for data preprocessing. In addition, we used the pgmpy 0.1.3 [24] library for the construction and evaluation of Bayesian Networks.

A. Dataset

In our experiments, we used a dataset provided by the Backblaze Company [25]. The Backblaze data comprise 59,654,783 daily observations of the state of several disks distributed among 93 different models collected between 04/10/2013 and 12/31/2016. These observations contain basic data for each device along with their SMART data. A device is considered failed when 1) it stops working or 2) it has shown signs that it will stop soon, such as, for example, its SMART self test fails.

Each data sample has serial number, model, capacity, label and 90 performance-monitoring attributes, that are the raw and normalized values for 45 different SMART stats reported by the drive. Most drives do not report values for all SMART attributes, so there are blank fields in every record. Also, different drives may report different statistics depending on their model and manufacturer.

From the Backblaze data, we selected the models that have more than 5 serials and 500 data records. The resulting dataset comprise 15 models described in Table I. In addition, we excluded those devices whose serial failed at some point and reappeared later. Table I shows the number of HDDs and the number of records for each model. The bold models were chosen as the targets for the execution of the proposed Transfer Learning strategies. We chose these three disk models considering their variety regarding lackness of data.

HDD Models	**Number of Serials**	**Data Records**
ST320005XXXX	6	511
ST250LT007	**7**	**2744**
ST3160316AS	8	2853
ST3160318AS	13	7044
ST32000542AS	13	2038
ST33000651AS	**16**	**3556**
ST1500DL003	35	8910
ST4000DX000	35	25658
ST6000DX000	42	10588
ST8000DM002	48	3531
ST320LT007	**80**	**54640**
ST31500341AS	102	22251
ST31500541AS	239	63161
ST3000DM001	1012	174474
ST4000DM000	1698	747296

TABLE I: HDD models with significant amount of data.

For the optimization of the Bayesian Networks construction process, we removed SMART attributes that do not have a strong impact in the failure prediction. We will use only the attributes selected in [7], which were the most important according to a feature selection process based on the Recursive Feature Elimination [26] using the Random Forest classifier. The selected attributes were: SMART 187 RAW, SMART 240 RAW, SMART 5 RAW, SMART 184 RAW, SMART 190 RAW, SMART 7 RAW, SMART 188 RAW, SMART 197 RAW that represent respectively Reported Uncorrectable Errors, Head Flying Hours, Reallocated Sectors Count, End-to-End error, Temperature Difference from 100, Seek Error Rate, Command Timeout and Current Pending Sector Count. Most of the selected attributes are in accordance with the attributes considered critical by the Backblaze Company.

B. Performance Evaluation

In our experiments we obtained results of all possible combinations of data source strategies and transfer learning methods. For each target HDD model, its dataset was splitted in 70% for training and 30% for testing the performance of the BNs. For the Linear and Logarithmic Aggregation methods, a grid search was performed to find the best weights, using a validation set of 20% from the training data. For the DBLP the hyperparameters were set as $c_{min} = 0.25$ and $c_{max} = 0.75$, as suggested in [16].

We chose target models with different number of records to verify the impact of such characteristic in our method. For Strategy 2, we used the disk models **ST3000DM001** and **ST4000DM000** to be the sources with more data. For Strategy 3, we made a clustered model source using the disks models **ST320005XXXX**, **ST32000542AS**, **ST1500DL003**, **ST31500341AS** and **ST31500541AS**.

The performances were assessed by three metrics: error mean, error median and accuracy. While accuracy is a commonly used classification metric, the error consists of the difference, in quarters of the year, between the classifier prediction and the disk actual remaining useful life. The error mean is important since small deviations in the classification are less severe than high deviations. Consider an example where the BN predicts that a given HDD has only two months left but the HDD still works for four months. This error is clearly less significant to quantify the performance of the model than a case where the model predicts ten months left for the same HDD. Although both models wrongly classified the same sample, the error magnitude is different. The third metric (error median) was chosen since the median is less sensitive to outliers. Table II shows the results for all target models, considering the aforementioned performance metrics.

According to the results, we can see that using any transfer learning strategy can improve the performance of the target model. This can be verified by comparing the results of each combination (source strategy + transfer learning method) to the "no transfer" approach. Another important aspect to be noticed is that, as expected, the performance gain is more significant for datasets with less records.

Comparing the combinations of transfer methods and source strategies we can verify that Strategy 3 combined with different transfer methods achieved the best results for all performance criteria on HDD **ST250LT007**. This is an important result since **ST250LT007** is the HDD with the lowest number of records and also the one that showed the highest performance gain when compared to the "no transfer" strategy.

In **ST33000651AS** and **ST320LT007**, we can verify that different combinations had the best performance. A possible explanation relies on the fact that these HDDs have more records and thus transfer learning may have less impact in the performance of each model. Although several combinations had the best results, it is noticeable that Strategy 3 either achieved the best results or had a similar performance to the best combinations.

VI. Conclusion

In this work we evaluated several transfer learning strategies to the task of failure prediction on Hard Disk Drives. The strategies tested combine three transfer learning methods and three approaches for building source datasets.

We used a failure prediction method based on a BN model and tested three well known transfer learning strategies for parameter transfer in BNs. In addition to two commonly used source building approaches, we proposed a new source building method called clustering-based information source. In the proposed method, we grouped several HDDs according to their similarity and built a novel information source for transfer learning.

On the basis of our experiments we can state that transfer learning methods can provide performance gains in HDD failure prediction models. This is even more noticeable for HDDs with reduced number of records. We also verified that the proposed source building method led to performance improvements and can be considered a valid alternative for transfer learning. Future works may include the investigation of semi supervised strategies for HDDs failure prediction.

Acknowledgment

This work was partially supported by Funcap and LSBD/UFC.

Results

Transfer Method	Source	ST250LT007			ST33000651AS			ST320LT007		
		Mean	Median	Accuracy	Mean	Median	Accuracy	Mean	Median	Accuracy
No transfer	-	2.3822	1.7765	49.817%	0.9929	**0.5245**	29.653%	1.9280	1.4362	25.099%
Linear aggregation	Strategy 1	1.7879	1.4577	47.996%	0.9344	0.7038	43.966%	1.8256	1.5504	24.794%
	Strategy 2	1.5973	1.5563	52.003%	**0.8458**	0.5617	42.469%	1.8214	1.5816	24.119%
	Strategy 3	1.4928	1.4570	**62.021%**	0.8769	0.5315	41.721%	1.8208	1.5407	24.000%
Logarithmic aggregation	Strategy 1	1.8677	1.2816	47.996%	0.9330	0.5386	37.792%	1.8496	1.5057	**28.720%**
	Strategy 2	1.6252	0.9721	50.091%	0.8736	0.5837	45.556%	1.8321	1.5254	26.350%
	Strategy 3	1.5563	0.9849	42.896%	0.8620	0.5650	45.556%	1.8250	**1.4279**	25.721%
Distance based linear pool	Strategy 1	1.6063	1.2528	56.102%	0.9386	0.7084	53.040%	1.7631	1.5487	23.814%
	Strategy 2	1.5632	0.9557	52.003%	0.9070	0.5690	**65.107%**	1.7778	1.5960	22.073%
	Strategy 3	**1.4207**	**0.8565**	56.102%	0.9103	0.6880	53.133%	**1.7554**	1.5329	23.556%

TABLE II: Results of the Transfer Learning strategies and methods combinations showing error mean, error median and accuracy.

References

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

[2] G. Vachtsevanos, F. L. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems: Methods and Case Studies*. John Wiley & Sons, 2006.

[3] 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.

[4] L. P. Queiroz, F. C. M. Rodrigues, J. P. P. Gomes, F. T. Brito, I. C. Brito, and J. C. Machado, "Fault detection in hard disk drives based on mixture of gaussians," in *2016 Brazilian Conference on Intelligent Systems (BRACIS)*, 2016.

[5] 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 Transactions on industrial informatics*, vol. 10, no. 1, pp. 419–430, 2014.

[6] 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.

[7] I. C. Chaves, M. R. P. de Paula, L. G. M. 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.

[8] 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.

[9] Y. Zhou, T. M. Hospedales, and N. Fenton, "When and where to transfer for bayesian network parameter learning," *Expert Systems with Applications*, vol. 55, pp. 361 – 373, 2016.

[10] P. Duygulu, D. Arifoglu, and M. Kalpakli, "Cross-document word matching for segmentation and retrieval of ottoman divans," *Pattern Analysis and Applications*, vol. 19, no. 3, pp. 647–663, 2016.

[11] Y. Yuan, X. Zheng, and X. Lu, "Hyperspectral image superresolution by transfer learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 5, pp. 1963–1974, May 2017.

[12] R. Li, Z. Zhao, X. Chen, J. Palicot, and H. Zhang, "Tact: A transfer actor-critic learning framework for energy saving in cellular radio access networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 2000–2011, April 2014.

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

[14] J. Pearl, "Probabilistic reasoning in intelligent systems," 1988.

[15] W. Lam and F. Bacchus, "Learning bayesian belief networks: An approach based on the mdl principle," *Computational intelligence*, vol. 10, no. 3, pp. 269–293, 1994.

[16] R. Luis, L. E. Sucar, and E. F. Morales, "Inductive transfer for learning bayesian networks," *Machine Learning*, vol. 79, no. 1, pp. 227–255, 2010.

[17] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, prediction, and search*. MIT press, 2000.

[18] D. P. Mesquita, J. P. Gomes, and L. R. Rodrigues, "K-means for datasets with missing attributes: Building soft constraints with observed and imputed values," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[19] P. Shopbell, M. Britton, and R. Ebert, "Making the most of missing values: object clustering with partial data in astronomy, astronomical data analysis software and system xiv," in *ASP Conference Series*, vol. 30, 2005, pp. 1297–1311.

[20] E. C. C. Jocelyn T. Chi and R. G. Baraniuk, "k-pod: A method for k-means clustering of missing data," *The American Statistician*, vol. 70, pp. 91–99, 2016.

[21] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[22] W. McKinney, "Pandas: a python data analysis library," 2008–, [Online; accessed 2017-04-26]. [Online]. Available: http://pandas.sourceforge.net

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] A. Ankan and A. Panda, "pgmpy: Probabilistic graphical models using python," in *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*, 2015.

[25] Backblaze. (2016, apr) Hard drive data and stats. [Online; accessed 2017-04-26]. [Online]. Available: https://www.backblaze.com/b2/hard-drive-test-data.html

[26] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1-3, pp. 389–422, 2002.