Pervasive and Mobile Computing 70 (2021) 101283

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Spatio-temporal AI inference engine for estimating hard disk reliability

Sanchita Basak [a,*], Saptarshi Sengupta [a], Shi-Jie Wen [b], Abhishek Dubey [a]

[a] Vanderbilt University, Department of EECS, Nashville, TN, USA
[b] Cisco, San Jose, CA, USA

ARTICLE INFO

Article history:
Received 13 February 2020
Received in revised form 15 July 2020
Accepted 22 October 2020
Available online 9 November 2020

Keywords:
Remaining useful life
Long short term memory
Prognostics
Predictive health maintenance
Hierarchical clustering

ABSTRACT

This paper focuses on building a spatio-temporal AI inference engine for estimating hard disk reliability. Most electronic systems such as hard disks routinely collect such reliability parameters in the field to monitor the health of the system. Changes in parameters as a function of time are monitored and any observed changes are compared with the known failure signatures. If the trajectory of the measured data matches that of a failure signature, operators are alerted to take corrective action. However, the interest of the operators lies in being able to identify the failures before they occur. The state of the art methodology including our prior work is to train machine learning models on temporal sequence data capturing the variations across multiple features and using it to predict the remaining useful life of the devices. However, as we show in this paper temporal prediction capability alone is not sufficient and can lead to low precision and the uncertainty around the prediction is very large. This is primarily due to the non-uniform progression of feature patterns over time. Our hypothesis is that the accuracy can be improved if we combine the temporal prediction methods with a spatial analysis that compares the value of key SMART features of the devices across similar model in a fixed time window (unlike the temporal method which uses the data from a single device and a much larger historical window). In this paper, we first describe both temporal and spatial approaches, describe the methods to select various hyperparameters, and then show a workflow to combine these two methodologies and provide comparative results. Our results illustrate that the average precision of temporal methods using long-short temporal memory networks to predict impending failures in the next ten days was 84 percent. To improve precision, we use the set of disks identified as potential failures and start applying spatial anomaly detection methods on those disks. This helps us remove the false positives from the temporal prediction results and provide a tighter bound on the set of disks with impending failure.

1. Introduction

Low-cost, high volume physical data storage devices have powered a deployment revolution in the consumer space for decades now. The deep reliance on information and data-driven technology which is now part of everyday life comes at the price of consistently meeting innovation and reliability challenges facing the data-storage systems: from those in deployment inside personal computers to large-scale industrial servers. Thus it is critically important to deepen

* Corresponding author.
E-mail address: sanchita.basak@vanderbilt.edu (S. Basak).

https://doi.org/10.1016/j.pmcj.2020.101283
1574-1192/

our understanding of how and why physical data-storage systems fail and what if any are its telltale signs. All of the manufacturing, installation, and maintenance policies may be well served by extending our insights on how to effectively monitor health statistics of such devices. However, hardware faults, in the absence of knowledge about the Remaining Useful Life (RUL), are commonly unavoidable and may cause violated service agreements on top of revenue losses [1]. If a disk is estimated to fail soon, it can be subjected to specialized, expert monitoring and the virtual machines can be allocated to other healthier disks and a backup process of the failure-prone disks can be started. This ensures higher availability and reliability of cloud-based services and reduces revenue costs associated with it.

Data from sensors onboard are typically exploited to perform prognostic work regarding RUL estimation or health checkups by scanning for anomalies or system-level faults. This can be performed using both model-driven or data-driven approaches. There have been several studies directed at predicting RUL of hard disks using model-driven approaches [2]. In such cases often a model using mathematical distributions is formulated, describing the behavior of device degradation. The model parameters are then estimated using measured data. Most electronic systems such as hard disks routinely collect such reliability parameters in the field to monitor the health of the system. In data-driven approaches, we do not have a physical model. Through observation of lots of data instances, we can apply machine learning and deep learning approaches to learn the pattern of the degradation model. However, model-driven approaches cannot always capture the device dynamics well using standard distributions especially when it is important to consider the trend of variation of model parameters across the devices [3]. This is primarily because as the number of variables and performance metrics required to predict future performance is very large it cannot be modeled through analytic means easily.

The primary mechanism of building data-driven RUL predictors is machine learning — with the underlying assumption being that any failures at the system level will be preceded by co-dependent changes in some of the parameters. These parameters are therefore monitored as a function of time and any observed changes are compared with the known failure signatures. If the trajectory of the measured data matches that of a failure signature, operators are alerted to take corrective action. However, there are problems with these approaches. In particular, these kinds of analyses usually assume that all deployed systems start with similar initial sensor value distribution – a highly unlikely scenario. Different disk models have their own initial set of parameters that are different from that of all other similar models. As a result, failure signatures for each model varies, even though the overall trend in failure signature remains the same. An AI engine (a data-driven RUL predictor and disk failure estimator) trained to notice such changes without understanding the baseline level will have many false positives and false negatives as part of the predictive routine. Such false failure predictions increase the cost by either attending to unexpected failures or by unnecessary repairs/replacements. To make the AI engine robust against wrong/missed inferences, vast data sets for training are needed, increasing the training cost and complexity of the inference engine. However, over time, storage and management of such large data sets becomes prohibitively expensive and complex.

Our Contributions: In this paper, which is an extension of our previous paper [4] we make three primary contributions. First, we show how we can train a recurrent neural network-based RUL prediction model by using a Deep Long Short Term Memory (LSTM) [5] on temporal sequence data capturing the variations across multiple features and using it to predict the remaining useful life of the devices. Specifically, we showcase a novel adaptive device-oriented normalization scheme to help with the variation and range of the S.M.A.R.T (Self-Monitoring, Analysis, and Reporting Technology) indices reported for all disks, reported once every twenty-four hours through the end of their reporting period (we use the Backblaze hard drive test data [6]. This is critical because the devices from a specific manufacturer do not necessarily fail with similar failure-specific feature values, or in other words, under similar conditions. Also, the concern lies in the fact that at some feature values flagged healthy for a device, another device from the same manufacturer may fail [7]. This makes the possibility of identifying a specific set of feature values pertaining to global failure a hard problem. Consequently, the feasibility of traditional Machine Learning (ML) approaches in learning from these highly non-linear causal embedding of features is put under question. The proposed work extracts the training data from such highly unorganized feature sets containing major class imbalances and establishes device-specific and customized normalization techniques in training and testing phases to mitigate this issue.

Second, we show that the temporal sequence data-based predictions uncertainty can be reduced by using data (from similar components) across all deployed systems in small and successive time windows. We call this *Spatial* analysis as it performs on a cluster of similar observations. If a disk is identified to fail within a few days according to the temporal analysis, then we run the spatial analysis some days before and after the predicted day of failure. For each such day, we only consider those disks that have a possible imminent failure suggested by temporal analysis and verify if they are listed in the set of outlier disks as given by the spatial analysis. The insight of combining these methods brought by this work is guided from the perspective that if according to the temporal analysis we had a false-positive decision about any disk, we now also can check if it does not show any outlier behavior according to the spatial analysis, thereby reducing the number of false positives. Also, if the temporal analysis identifies a disk to be falsely negative, i.e., it predicts that the device is not going to fail soon whereas actually the device has an impending failure, the spatial analysis can still identify it as an outlier so that the disk can be monitored for possible failure. Thus, by considering both spatial and temporal analysis outcomes for any particular disk before isolating it as a probable candidate for failure we limit the number of suspected failures or false positives.

Third, to validate the results we show a novel methodology, that extracts a *simulation* subset of the data which have not been used in *training* or *validation* purposes. Because during training, the device failure logs are already known which

can be used in preprocessing the data, but in real-world scenarios, the Remaining Useful Life (RUL) of devices under simulation set is not known, and there comes the challenge to preprocess the simulation data in a way so as to have similar inherent feature mappings as that of the training set, thus ensuring the extension of the prediction capacity to the simulation data.

Our results show the reduction of Mean Absolute Error (MAE) of RUL prediction from 5 days to 2.4 days when the spatio-temporal analysis is applied instead of temporal analysis. The outline of the rest of this paper is as follows: Section 2 discusses the prognostic problem and related research, Section 3 discusses our approach, Section 4 walks through the experimental results, Section 5 concludes the paper.

2. Understanding the prognostics problem and related research

The generalized problem that we are solving is to improve the device health prediction mechanism using both spatial and temporal analysis. In case of spatial analysis, the problem is to identify a set of disks that show anomalous behavior with respect to the entire set of disks. In case of temporal analysis, we are trying to formulate an effective normalization strategy. We have complete logs of data for devices under training including the start of data collection up to their failures. To validate our approach, we simulate a real-time scenario where the data is coming in real time, providing access to the data from the start of data collection up to the current day when the device is being tested. Note that the normalization strategy during the simulation is complicated because we the feature values at which different devices fail vary drastically. Thus, a related problem for us is to normalize the online simulation data in a way such that they have alike or close inherent feature patterns as that of the training data. Only then the trained model is able to predict the remaining useful life for the devices under the simulation set.

Note that the underlying motivation for pursuing system health monitoring is that if adequate structured information is obtained from historical or real-time data, efficient prediction models may be proposed by learning the evolution of implicit state variables. Such efforts seek to close the knowledge gap in advance about the timestamp in the life-cycle of a product after which it experiences complete operational failure or does not perform at its optimum level anymore. This is often referred to as the Prognostics problem.

2.1. Related research

Prognostic techniques are categorized broadly into: (i) Model-driven and (ii) Data-driven approaches. Traditionally model-driven approaches rely on modeling the failure distribution with statistical formulations. Gaussian Processes, Poisson distribution, Weibull distribution and Survival analysis methods have been widely used as they do a good job of learning patterns in low dimensional data and are limited by their reliance on state evolution equations put forth by assuming domain-specific knowledge. However, this has been known to be hard for modeling the reliability of hard disks. For example, B. Schroeder et al. [2], showed in their studies that Poisson distribution does not provide a good fit for the number of disk replacements, rather Gamma and Weibull distributions can capture the distribution of time between failure. However, Wang et al. [3], contended this fact and showed that time between failure is hard to model with a well-known distribution. None of the distributions including exponential, Weibull, gamma and lognormal fits the time-between-failure data. On the other hand there has been some success in model based methods as shown by Hu and Liu [8]. They used a state space model (SSM) which is a sequential Monte-Carlo approach that uses Bayesian estimation to assess the remaining useful life of devices.

As the problem complexity increases, model-driven approaches cannot keep up with learning the inherent causal embedding in the data as well. Priors and updates used in these kinds of models are hand tuned and simple, and are unable to capture the intricacies and functional relationships in high dimensional training data. Self-organizing representation learning disciplines such as deep neural networks reduce the roadblock to structure learning from voluminous sensor data and do not necessitate hand-crafting state equations of system evolution. Given that accurate labels are annotated for the training data, data-driven methods may be applied to complex prediction tasks with an accuracy unmatched by simpler, model-driven approaches.

Therefore, recent works on prognostics and device health management have primarily used data-driven approaches. Eker et al. [14] carried out RUL prediction comparing sensor data similarities tested on three datasets on various systems undergoing degradation. Deep convolutional neural network (CNN) based RUL prediction studies were done by Sateesh Babu et al. [15]. Recurrent neural network (RNN) studies were done by O Heimes [16]. Gugulothu et al. [10] proposed a novel approach known as 'Embed-RUL' that does not rely on assumptions about degradation trends but uses time series embeddings based upon recurrent neural network. Recent works like [17] also emphasize the use of recurrent neural networks to model inherent patterns of sensor observations that dynamically varies across time cycles. The authors in [18], proposed strategies for directly identifying the healthy and erroneous state of devices using disk replacement prediction algorithm with change-point detection and tested their approach on Backblaze data. On the other hand, Aussel et al. [7] applied SVM, RF and GBT on the same dataset to predict hard disk failures and reported the corresponding precision and recall.

Various swarm intelligence algorithms [19] including quantum particle swarm optimization [20,21] have also been used in predicting device health as evidenced by the work of Yu et al. [22] who carried out remaining useful life prediction

Table 1
Approaches for hard disk prognostics.

Approach	Paper	Contribution
Model-driven	Schroeder and Gibson [9]	Established that the distribution of failure instances in hard disks can be approximated by Gamma and Weibull distributions
Model-driven	Wang et al. [3] [9]	Showed that the time-between-failure (TBF) distributions are difficult to model using commonly used statistical distributions.
Data-driven	Gugulothu et al. [10]	Put forward a prediction scheme by using a recurrent neural network model and generating embeddings that capture multivariate time series data trends.
Data-driven	Malhotra et al. [11]	Introduced EncDec-AD (Encoder–Decoder scheme for Anomaly Detection) which seeks to reconstruct normal time-series behavior and subsequently uses reconstruction error for anomaly detection. This reinforces the idea of using RNNs to capture intricate dependencies among sensor observations in cases where external factors may render time series observations to be unpredictable.
Data-driven	Botezatu et al. [12]	Proposed an analysis pipeline using SMART indices which can efficiently predict disk replacement necessities 10–15 days in advance over 30000 disks from two major manufacturers, observed over 17 months of time.
Data-driven	Aussel et al. [7]	Used the same dataset (Backblaze) in estimating hard disk failures using SVM, RF and GBT factoring into account the highly unbalanced nature of the data. They reported a precision of 95% and a recall of 67% on a one year sample of over 12million examples with only 2586 failure instances.
Data-driven	Li et al. [13]	Discussed various data-driven mechanisms in estimating health and lifetime of lithium ion batteries. They outline in detail the analytical models based mostly on machine learning strategies that are used in various works for diagnosis as well as prognostics purposes. They emphasize on the differential analysis methods that in general can be used to identify signatures of device aging across various systems.

of lithium-ion batteries. Aussel et al. [7] used random forest decision trees to predict remain useful life. However, their method is more closely related to our spatial analysis method rather than the other previous hard disk reliability methods that use temporal sequences. The Table 1 summarizes various approaches using model-driven and data-driven techniques for prognosticating device health.

In contrast, we rely on spatio-temporal analysis of device health monitoring. Spatio-temporal grouping of events reveal patterns and understanding of data distributed in a multi-dimensional space. For example, emerging spatio-temporal analysis in traffic data helps identify as well as prognosticate traffic propagation patterns under congestion as evidenced in [23].

3. Our approach

The overall workflow that summarizes the proposed spatio-temporal analysis is described in Fig. 1. We show the temporal analysis training workflow, temporal analysis inference workflow, spatial model construction workflow and the combined spatio-temporal inference workflow. We discuss these approaches below in subsequent subsections. However, we first describe the dataset we use.

3.1. Dataset and SMART features

The Backblaze hard drive dataset [6] contains information about more than 90,000 hard disks from various manufacturers. For each device the records include some basic drive information such as date of the record, model and serial numbers and a field indicating the status of the disk whether it has failed or is active as well as various S.M.A.R.T parameters. These S.M.A.R.T (Self Monitoring Analysis and Reporting Technology) parameters indicate various information on drive usage and error profiles required to monitor the device health. We work with the data from 2017 which reports a total of 40 S.M.A.R.T statistics for each drive, though all the parameters are not reported by each manufacturer. We chose to work with Seagate drives due to their high failure instances in the past. Specifically we work with ST400DM000 as it contributed to most of the failures among all the models from Seagate. Seagate records twenty four S.M.A.R.T. features out of which we selected five features to work with along the line of our work in the past [4]. The temporal progression

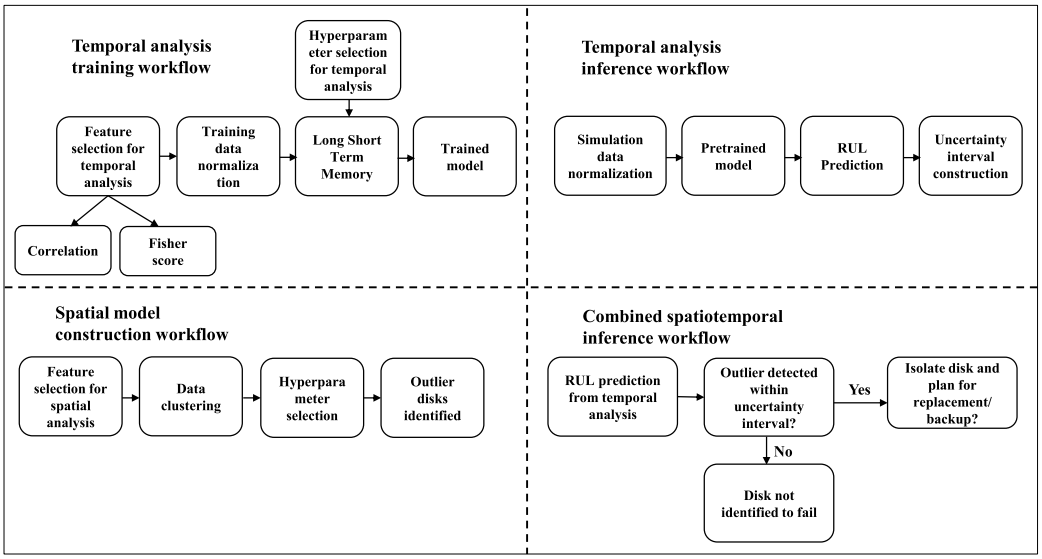

Fig. 1. Workflow for applying the spatio-temporal analysis.

Table 2
Summary of SMART features used for analysis.

No.	Analysis mode	SMART ID	Attribute name	Description
1	Spatial	SMART 1	Raw read error rate (normalized value)	Rate of hardware error while reading data from a disk surface
5	Temporal	SMART 7	Seek error rate (raw value)	Frequency of the errors during disk head positioning and increases as the device approaches failure.
6	Temporal	SMART 9	Power-on-hours count (raw value)	Estimated remaining life of a device, based on the time a device was powered on.
9	Spatial	SMART 183	Runtime bad block (raw value)	Number of data blocks with uncorrectable errors
11	Spatial	SMART 187	Reported uncorrect (raw value)	Number of unrecovered errors
12	Spatial	SMART 188	Command timeout (raw value)	Number of aborted operations due to HDD timeout
19	Spatial	SMART 197	Current pending sector (raw value)	Number of unstable sectors
21	Spatial	SMART 199	Offline uncorrectable (raw value)	Number of errors in data transfer through inference cable
22	Temporal	SMART 240	Head flying hours /transfer error-rate (raw value)	Time spent for positioning of the drive heads
23	Temporal	SMART 241	Total LBAs written (raw value)	Related to the usage and aging process of devices
24	Temporal	SMART 242	Total LBAs read (raw value)	Related to the usage and aging process of devices

of these features exhibit higher correlation with failure. It is to be noted that these selected features have a monotonically increasing trend, but their ranges vary vastly among devices even from the same manufacturer to the extent that both active as well as failure prone devices might exhibit similar feature values. So there is no fixed set of features that uniquely identifies failure prone situations.

We divide the data into training and validation sets. The training and validation set contains data where we know the ground truth, i.e. the time and feature corresponding to the failure. This information is used in data preprocessing

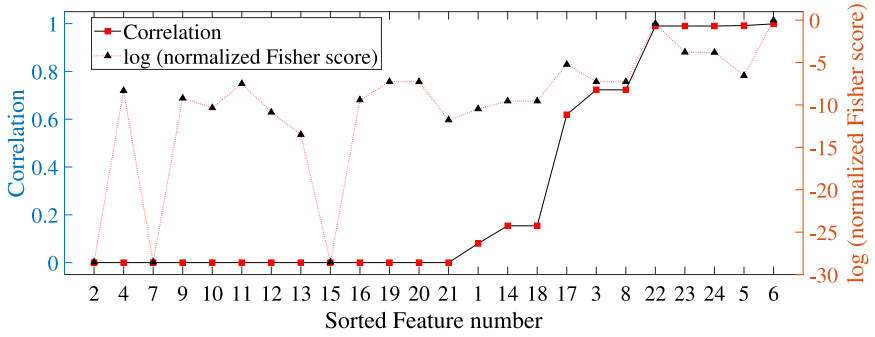

Fig. 2. Sorted feature values in ascending order of correlation and Fisher score.

that will be discussed in the subsequent sections. The test dataset contains devices that are active now but may fail soon. The day of failure for the test devices are unknown. Hence the test data cannot be preprocessed in the same manner as that of the training data. Now in order to apply the trained model to the test data, the test data should be adjusted and processed in a way so that it has similar or close pattern of distribution of features as that of training data. This makes the problem challenging as there is no global set of features w.r.t. which this normalization can be done.

Out of this whole dataset chose to work with features listed in Table 2 that are correlated most with the failure. Note that out of this different subsets show stronger correlation across spatial and temporal dimensions. We indicate that in a separate column in the table. We collected the data for all Seagate ST4000DM000 models that failed in June and July, 2017. Thus the training dataset contains temporal data sequences for each such device up to the day of failure. In the combined training dataset, we search for unique serial numbers of the devices and the data associated with any particular serial number is normalized with respect to the features corresponding to the day that device failed. This normalized training dataset has the label of Remaining Useful Life (RUL) associated with each instance of data. The RUL associated with the day of failure is zero and increases by 1 as we look back the data for the past days one-by-one. So the normalized training dataset contains multiple instances of data of different devices from the same manufacturer, but associated with similar remaining lives.

3.2. Temporal prediction analysis

We describe the feature selection, normalization and training approach for the temporal analysis workflow in this section.

3.2.1. Feature selection for temporal analysis

For temporal analysis, we chose to work with five features as listed in Table 2 that are correlated most with the temporal variation of failure. The feature set includes SMART 7 (raw value), SMART 9 (raw value), SMART 240 (raw value), SMART 241 (raw value) and SMART 242 (raw value). To select these features, we carried out both correlation and Fisher score analysis. Then, the features were sorted according to their individual correlation with the remaining useful life and the features with highest value of correlation score were selected.

In Fig. 2 we present the correlation of each feature with failure with the features sorted according to increasing order of their corresponding correlation values. We also calculate and plot the logarithm of normalized Fisher score [24] values of each feature of the features in Fig. 2. It prioritizes features having better distinguishing capability i.e., larger variance of values among separate classes and more similar values within the same class.

We selected the five features that have the highest correlation with failure and the corresponding Fisher score is also high. After the first five feature with highest correlations, the correlation score drops significantly by more than 20%. Including features that are not directly tied to failure will hinder the neural network training as it will not be able to distinctively identify features that are correlated to failures and that is why we restricted our analysis with those features that are mostly correlated.

As we worked with this data we needed to arrange it as a three dimensional matrix to be fed at the input of deep LSTM networks. A brief overview of LSTM architecture we used is given in Section 3.2.3. An LSTM network accepts input in the form of number of instances $\times$ number of time steps to look back $\times$ number of features (Section 3.2.3). For each training instance we look back at twenty five days of data to predict the RUL of a device evaluated on the test day. This look back sequence varies according to the problem at hand. In this case, this hyperparameter value has been chosen as it results in the least validation loss for the chosen data.

As discussed earlier, the training and test data pre-processing are different given the RUL estimation approach developed in this work. In order to apply the trained network to the test dataset, the test data distribution should be similar to that of training data distribution. The problem is that for the training data, the data for each device is

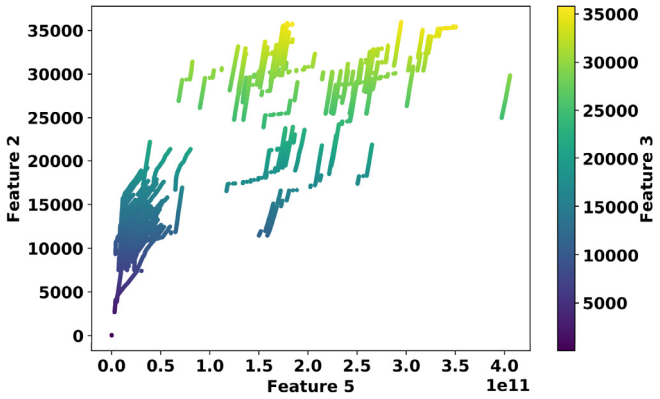

Fig. 3. Variation of feature 2 vs. feature 5 color-coded by feature 3.

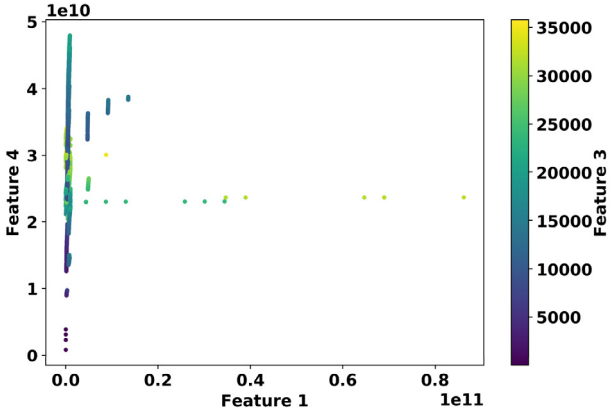

Fig. 4. Variation of feature 4 vs. feature 1 color-coded by feature 3.

normalized with respect to the features corresponding to the day of failure. But for the test set, where the devices are active at present but are going to fail sometime soon, the data cannot be normalized in a similar way. This is because the features corresponding to the failures are unknown. To overcome this roadblock, we propose a feature-specific, adaptive normalization approach for the test data which we will discuss next.

3.2.2. Normalization of simulated test data

As discussed earlier, the range of data used for training vary vastly among devices even though all the devices under training are close to failure. To illustrate this issue further, Fig. 3 shows the scatter plot of feature 2 (SMART 9) vs. feature 5 (SMART 242) color-coded with respect to feature 3 (SMART 240). It shows that feature 2 varies from almost 0 to 35000 whereas feature 5 varies from 0 to $4e^{11}$ for the set of training devices. The scatter plot displays multiple groups or chunks of data made of comparatively similar devices in terms of their feature ranges before failure. The variation of feature 2 vs. 5 is color-coded by the variation in feature 3 which also shows different color patterns at different data chunks. Similarly, Fig. 4 shows the scatter plot of feature 4 (SMART 241) vs. feature 1 (SMART 7) color-coded by feature 3 (SMART 240). This presents different grouping patterns of the features with distinct dense and sparse distributions. These differences in grouping patterns or densities calls for careful choice of normalization threshold for which we also need to take a look at the historical data distribution.

To normalize the data, we start by looking into two months of data for each device that we work with. We found that although there are fluctuations in feature values within shorter span of time, the overarching pattern of moving average has an uptrend over months. The interesting fact is that the distribution of each feature remains mostly similar in a span of two months (Fig. 5). Thus consideration of past two months of data in fetching historical maximum is appropriate in the sense that the time period is not too short to be influenced largely by noisy fluctuations of short span of data sequences and not too long and thus balances the computational overhead. Given the selected data we try two different normalization strategies.

- **Max value normalization (Strategy 1)**: We normalize the features with respect to the historically maximum feature value. We assume the devices will last till the maximum attainable value of features are met given the past

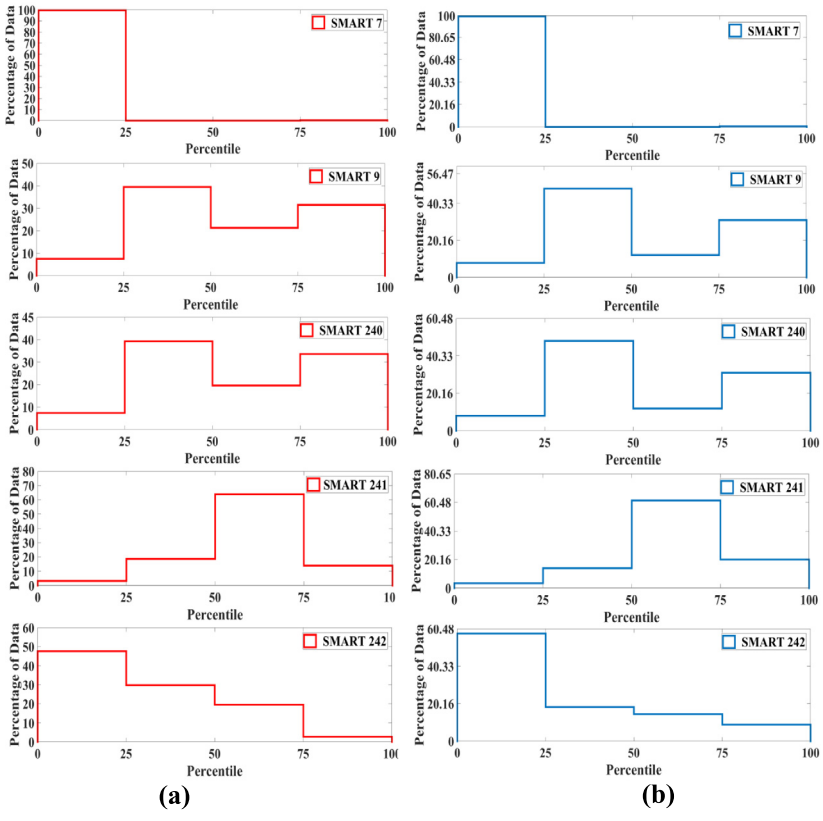

Fig. 5. The histogram indicates that the distribution of features selected for temporal analysis for any two consecutive months are similar.

observation. However, note that there is a possibility that maximum value is an outlier and normalization with respect to that will result in lower fractional values resulting in larger RUL values than actual.

- (75*th* **percentile) (Strategy 2)**: Thus, we also consider a second strategy where we use the supremum of 3rd quartile (75*th* percentile) of the sorted data as the normalization threshold. The motivation for this is driven by the fact that most of the data is contained within this bound of the distribution. To keep the normalization threshold generic to be applied to any time of the year we restrict our granularity at the level of quartiles. The 75*th* percentile threshold is optimal in this case as it is not too far from the historical maximum containing most of the data samples but mostly excludes the outliers in the distribution.

3.2.3. LSTM network

In order to capture the temporal progression of various device metrics towards failure we use Long Short Term Memory networks (LSTM) [5]. LSTMs mitigate the vanishing gradient problem that appears during backpropagation in Recurrent Neural Networks for processing long data sequences hindering the learning process. Although LSTMs have similar control flow of information as that of RNNs, each LSTM cell is equipped with forgetting or memorizing relevant information and contains input, output and forget gate. The sigmoid activation function controlled forget gate captures both the current input and previous hidden state information to decide upon which information is to be kept or forgotten. The input gate is controlled by sigmoid and tanh activation function to update the cell state by considering relevant information from the current state. The output gate determines the next hidden state. The equations governing the workflow of an LSTM unit are as follows:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2}$$

$$c_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * c_t \tag{4}$$

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

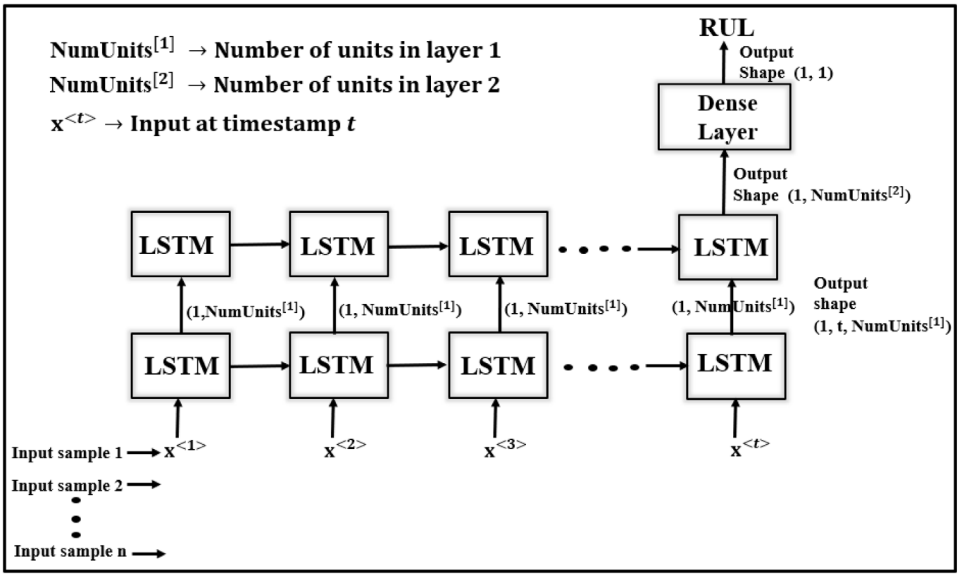

Fig. 6. The LSTM architecture used in this work.

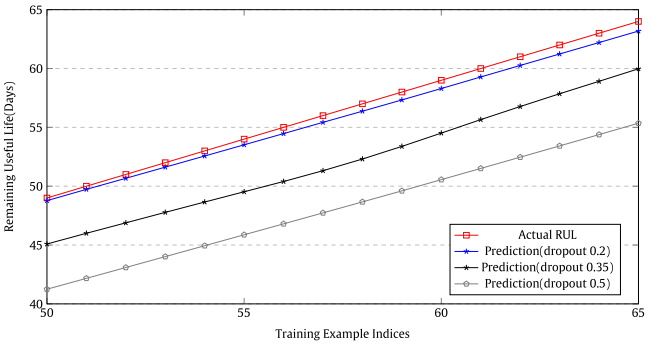

Fig. 7. Selection of optimal dropout ratio.

The input gate i_t, forget gate f_t and output gate o_t are responsible for determining the value of the cell state C_t. The forget gate layer takes the output of previous timestep h_{t-1}, the current input x_t and applies 'sigmoid' activation function to output a value between 0 and 1 for each value in the cell state C_{t-1} denoting the probability of keeping the previous information. The input gate layer i_t, which is also a sigmoid activation layer, decides which of the values to be updated. The new cell state C_t is influenced by the previous state C_{t-1} as well as i_t, f_t and h_{t-1}. A sigmoid output layer o_t decides upon what parts of the cell state to be outputted. The hyperbolic tangent activated cell state multiplied with the output gate o_t produces the output of the LSTM block at current time step.

To train the Remaining Useful Life prediction model we use a bilayered LSTM architecture with 100 units in each layer. We select RMSprop as the optimizer and use dropout to control overfitting. Fig. 6 shows the LSTM network used for this problem unfolded in time. Temporal sequences of training instances are fed to the network to predict the RUL of devices. The figure lists the shape of the outputs produced by each LSTM cell.

During the training we had to select a number of hyperparameters as well. We discuss them below.

1. **Dropout Ratio:** Very high values of dropout ratio may shut down most of the units of hidden layers resulting into non-optimal decision boundaries. On the other hand, if the dropout ratio is very low for all the layers, then the neural network leads to overfitting of the training data and greater error in validation or simulation set. So, a moderate dropout ratio gives optimal or near optimal decision boundary. The Fig. 7 shows the influence of dropout ratio on the accuracy of RUL prediction. A dropout ratio of 0.2 is chosen for both the layers according to our findings.
2. **Units in each layer:** The Fig. 8 shows a graph showing the influence of variation of number of units per layer on the training and validation loss as well as the execution time. It is evidenced that the optimal number of units per layer is 100 in this case with moderate training loss, lowest validation loss and quite low execution time which largely increases with increasing number of units.

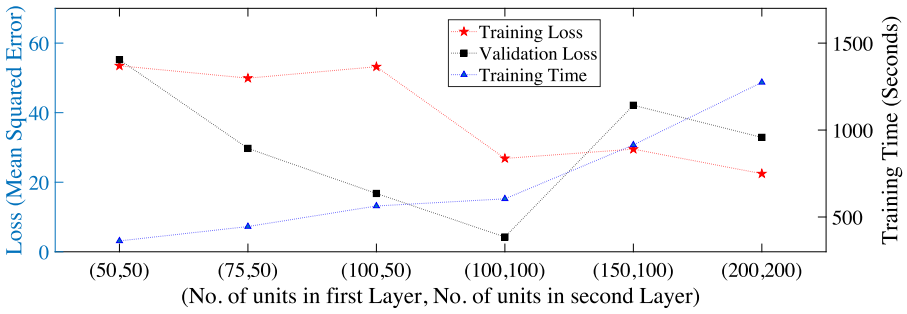

Fig. 8. Variation of training loss, validation loss and training time with number of units in each layer.

3. **Data normalization length:** We wanted to predict failure of a device at most 125 days in advance, that is why we chose the parameter value as 150 which is the number of days we look back from the current day while performing the temporal analysis. The LSTM takes the normalization length as 25, that is why the parameter value as 150 so that even if we want to predict for failure 125 days prior to actual failure we will need data up to 125+25=150 days back. Now, normalization length of 200 will enable prediction capabilities to identify failures 175 ($200-25=175$) days in advance, but the predictions will be much weaker.
4. **Range of historical data:** We look ed at two months of past data to study the data distributions. Also taking any other range of past data instead of two is completely user-defined. Just the normalization threshold will differ which needs to be studied from the data distributions of the corresponding time range.

3.2.4. Using RUL prediction to predict failures

Note that the remaining useful life prediction at any time step provides the estimate of reliability of the device. If the RUL is less than a prespecified threshold the data center operators can choose to replace the device. However, as we know the machine learning based models have an inherent uncertainty and there is always an error and as such we can estimate an uncertainty bound around the predicted RUL results. We will discuss the specific uncertainty bound that we determined for Backblaze datasets in the results section (Section 4).

3.3. Spatial analysis

For each day we have access to the SMART statistics of almost 34500 Seagate ST4000DM000 hard disks. Our goal is to identify the outlier disks through spatial analysis that are going to fail on that day based on a two-step hierarchical clustering.

3.3.1. Feature selection for spatial analysis

Unlike temporal analysis, there are existing recommendations of features (SMART 5, 187, 188, 197, and 198) (Table 2) that should be used for spatial clustering, as these features show the best daily correlations [25]. We start with these features and check for the percentage of failure cases where these feature values are greater than zero for Seagate ST4000DM000 devices. We repeat the procedure for 14 consecutive days where there was at least one failure of this particular device model. The Fig. 9 shows the percentage of failure-related cases when each of these five suggested features had their error counts greater than zero. We see that all the failures cannot be explained by any single SMART feature. On average, at the time of device failures, SMART 5 and SMART 188 were only 19.14% and 6.5% of the times greater than zero. So they were eliminated from the first phase of hierarchical clustering. SMART 187, 197, and 198 had there error count greater than zero for 41.42%, 45.64%, and 45.64% times respectively. Backblaze also suggests that SMART 197 and SMART 198 have very good correlation such that they can be considered as the same index [26]. So we only took into account SMART 187 and 197 for the first phase of clustering. We also noticed that not only the value of the features that are correlated with failure but the change in the value of the features are important to track. So we used the change in values of SMART 187 and 197 as the features for the first phase of the spatial analysis.

3.3.2. Data clustering

We follow a two step isolation approach. In the first step, we specifically monitor SMART 187 (raw) which indicates reported uncorrected errors and SMART 197 (raw) indicating current pending sector count. For all the Seagate ST4000DM000 hard disks, we monitor the change in these two error profiles from past four days up to the current day. The choice of the number of days to monitor the changes in SMART values will be discussed in Section 3.3.3.

We track the changes of these two features denoted as $S187_{change}$ and $S197_{change}$ for all the disks and use hierarchical clustering [27] to cluster them. We consider the group with maximum number of elements as normal set of devices and

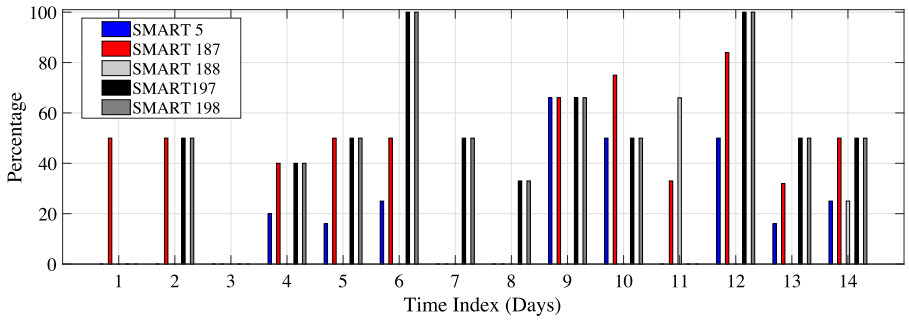

Fig. 9. Selection of features for spatial clustering. The plot shows which of the SMART features show variation in their values when the devices fail.

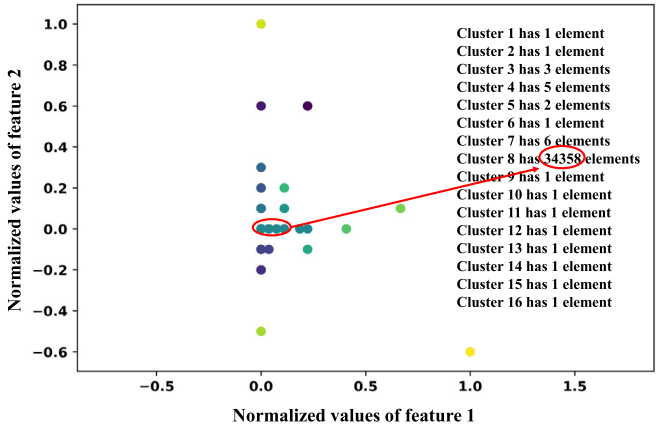

Fig. 10. Spatial Clustering of hard disks to determine outliers.

the rest of the clusters are considered as outliers. Fig. 10 shows an example where out of 34385 Seagate ST4000DM000 hard disks at a particular day, 34358 disks fell into normal category and the rest are considered as outliers.

The goal of the spatial analysis is to improve the recall (based on the RUL prediction from the temporal analysis), i.e. to lower the value of false negatives. In order to improve the precision we will combine the temporal analysis discussed later in this section. To improve the recall value as achieved by the first step of spatial analysis, i.e. to identify all or maximum number of disk failures that happens each day, we go for a second step of disk isolation. Here, we monitor the SMART parameters including SMART 1 (normalized), SMART 183 (raw), SMART 199 (raw), SMART 188 (raw), SMART 187 (raw) and SMART 197 (raw). We considered only the current value of all the features for any day and came up with a single feature value ($f_{combined}$) that is a weighted sum of the features discussed above.

One point to note that, Botezatu et al. [18] suggested that for Seagate devices the SMART 1 normalized value great than 117 indicates the necessity of disk replacement on the day. This threshold of SMART 1 normalized value was congruent with our observations. That is why we chose this specific threshold for SMART 1. Based on our observation, we also found that other than SMART 1 (normalized), SMART 183 (raw), SMART 199 (raw) and SMART 188 (raw) indicate a failure when they cross the zero error count. To compute the spatial analysis score, for each device, we initial set the feature $f_{combined}$ to zero. If the value for SMART 1 is greater than 117 or any of SMART 183, SMART 199, SMART 188, SMART 187 and SMART 197 have a value greater than zero, then or each such case, $f_{combined}$ is increased by 1 giving all the cases equal importance. Thereafter, we take all three features $S187_{change}$, $S197_{change}$, and $f_{combined}$ for all the ST4000DM000 disks and use hierarchical clustering to cluster them. Then the cluster with less number of members can be marked as anomalous. Note that while this approach improves the recall, the number of false positives that fall in the outlier increases. However, if we restrict the analysis set to only the devices whose RUL predicted by the temporal analysis is below a limit then we reduce the number of false positives. We will discuss this further when describing the results.

3.3.3. Hyperparameter selection for spatial analysis

For spatial analysis, the choice of the number of days to monitor the changes in feature values is an important hyperparameter. From current day, we choose to go back up to four days instead of just the day before, because in many cases, the failing devices have their error profiles changed largely not just between the day of failure and the previous day, but also between two to three days prior to the failure day. Also, they may have no changes in the error profile just

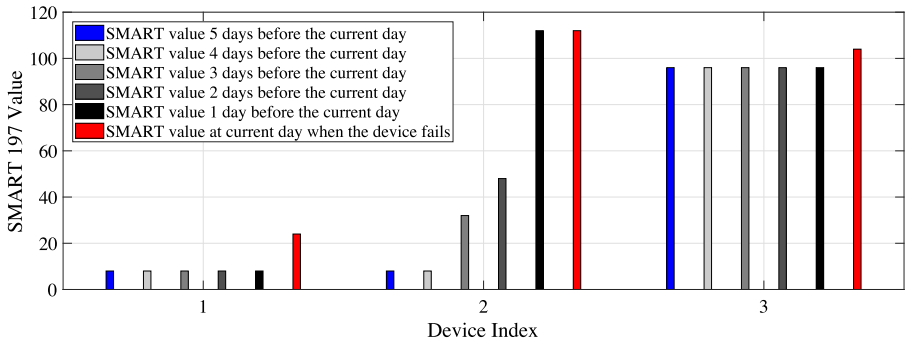

Fig. 11. Selection of optimal number of days to look back from the current day for spatial analysis.

before the day of failure. So to capture the effects of all the hard disks that have variations of feature values in the last few days, we empirically arrived at the decision of looking back for the past four days.

For example, on August 9, 2017, three Seagate ST4000DM000 devices failed as indicated by the device indices in Fig. 11. The plot shows SMART 197 value for all these devices for the day when the device failed as well as one, two, three, four, and five days prior to that day. It is observed that for devices 1 and 3, there is a jump in SMART 197 value between the day of failure and the previous day and there was no change of value from the past five days to the past one day. But for device 2, interestingly there was no change of SMART 197 value between the day of failure and the previous day. So if we had looked at SMART parameter changes only between the day of failure and the previous day, we would have missed marking this device showing possible outlier behavior. The SMART parameter change occurred between 4th and 3rd day, 3rd and 2nd day as well as 2nd and 1st day prior to failure and no change was observed between 5th and 4th day prior to failure. So if we look back past four days from the day of failure we have a better chance to identify outlier disks as the difference in SMART values between the day of failure and four days prior to that will incorporate any changes in SMART profile between any pair of days within that range. Now we can look at SMART parameter changes occurred more than four days prior to the current day when a device is being tested for. That is a user-defined hyper-parameter and can be varied. For our case, we observed that in most of the cases, the fluctuation in SMART parameters can be captured if we go back up to four days. That is why we chose to monitor error profiles up to four days back.

3.4. The combined spatio temporal analysis workflow

Recall that the problem we aim to solve is to improve our device health prediction mechanism using both *spatial* and *temporal* analyses. Observing the evolution of features with respect to time for a particular device is denoted as temporal approach whereas observing the entire group behavior of all the devices at a particular time is denoted as the spatial approach as it covers the space or expanse of the total set of devices at once. Note that in traditional spatial analysis, the geographic proximity or distance plays a major role. But in this case, the concept of distance among the devices is analogous to the euclidean distance of their features in the entire space of all the devices from the same manufacturer and model and we observe their group behavior as a collection of clusters in multidimensional space.

Fig. 12 shows an example of device health prediction for a disk i using spatio-temporal analysis. We feed the temporal sequence of feature set to the pre-trained LSTM model to predict its remaining useful life. If the device is predicted to fail after k days, we assume that the device will fail within $(k-d)$th day to $(k+d)$th day from the current day, where d is the confidence interval of the RUL prediction as discussed in Section 3.2.4. Then, we run the spatial analysis from $(k-d)$th day to $(k+d)$th day from the current day to check if the device falls into the outlier category. The example in the figure shows that the device i falls into the set of normal devices during $(k-d)$th day to $(k-2)$th day. It moves to the outlier category on $(k-1)$th day which indicates that the device is supposed to fail on $(k-1)$th day or very soon. Note that in most of the cases a disk may start showing its outlier/failure-prone behavior a few days before the actual failure. If the device is not removed on $(k-1)$th day, then it may fall into the outlier category on (k)th day and actually fails on that day. So from $(k+1)$th day to $(k+d)$th day the device is not available for testing as it has already failed and is then removed from the device list.

Thus the problem that we aim to solve is to improve our device health prediction mechanism using both *spatial* and *temporal* analyses. Observing the evolution of features with respect to time for a particular device is denoted as temporal approach whereas observing the entire group behavior of all the devices at a particular time is denoted as the spatial approach as it covers the space or expanse of the total set of devices at once. In traditional spatial analysis, the geographic proximity or distance plays a major role. But in this case, the concept of distance among the devices is analogous to the euclidean distance of their features in the entire space of all the devices from the same manufacturer and model and we observe their group behavior as a collection of clusters in multidimensional space. That is why we termed it as spatial analysis. So in spatio-temporal analysis we tried to emphasize two distinct branches of approach, one looking at the entire space at a single time and the other looking at the time series progression of individual devices.

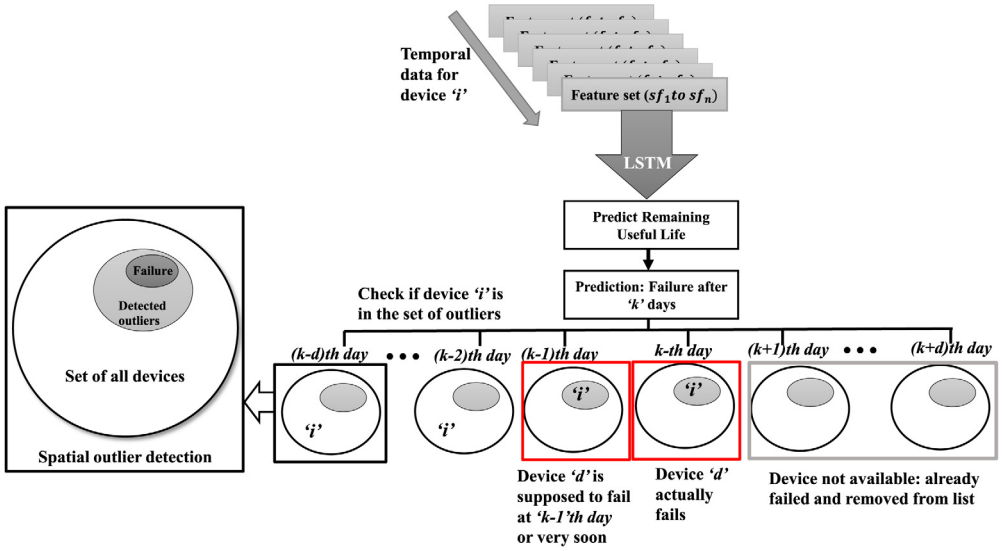

Fig. 12. General overview of our approach: the example describes that according to the temporal analysis, if a disk 'i' is predicted to fail after 'k' days from current time, a spatial analysis is carried out from 'k-d' to 'k+d' days to check if the device falls into the outlier category.

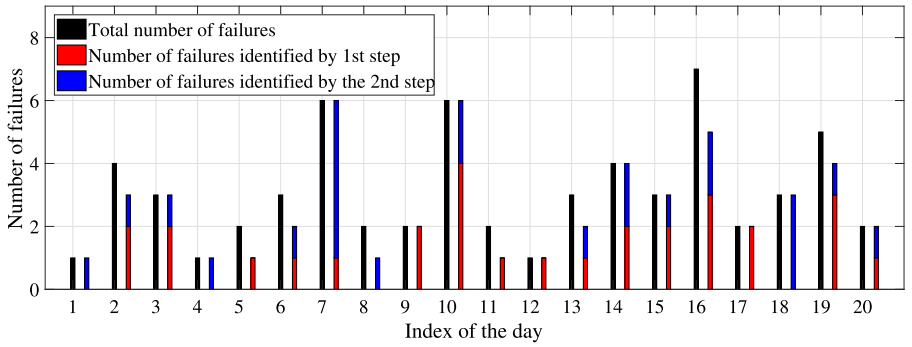

Fig. 13. Number of disk failures identified by the first and second step of spatial analysis compared to the total number of actual failures on each day.

4. Results and discussion

4.1. Spatial analysis results

At first we solely performed the spatial analysis for twenty days randomly selected from May to September, 2017. Fig. 13 shows the total number of disk failures on each day as well as the number of disk failures identified by the first and second step of spatial analysis. This also indicates the importance for the second step of spatial clustering as some of the faulty devices could not be identified by the first step. It is seen that in most of the cases all or in some cases most of the failures can be identified by the first and second step of spatial analysis.

Table 3 shows the precision and recall for identifying faulty disks solely by running spatial analysis for twelve consecutive days. When a disk failure is identified by the spatial analysis and the disk actually fails, we consider it as true positive. When a disk is identified to fail as per the spatial analysis but does not actually fail, we consider it as false positive. When a disk actually fails, but cannot be identified by the spatial analysis, we consider it as false negative. From the true positive, false positive and false negatives we calculate the corresponding precision and recall in identifying disk failures solely using spatial analysis. The results indicate that the recall values are high, but the corresponding precision values are very low. The average recall is 0.87 and the average precision is 0.00018. Note that this is expected because spatial analysis alone is not very accurate.

Table 3
Precision and recall values for spatial analysis.

Day index	Precision	Recall
1	0.00018	0.75
2	0.00006	0.5
3	0.00006	1
4	0.00024	1
5	0.00018	1
6	0.00012	1
7	0.00031	0.714
8	0.0003	1
9	0.00012	0.66
10	0.003	0.833
11	0.00018	1
12	0.00012	1

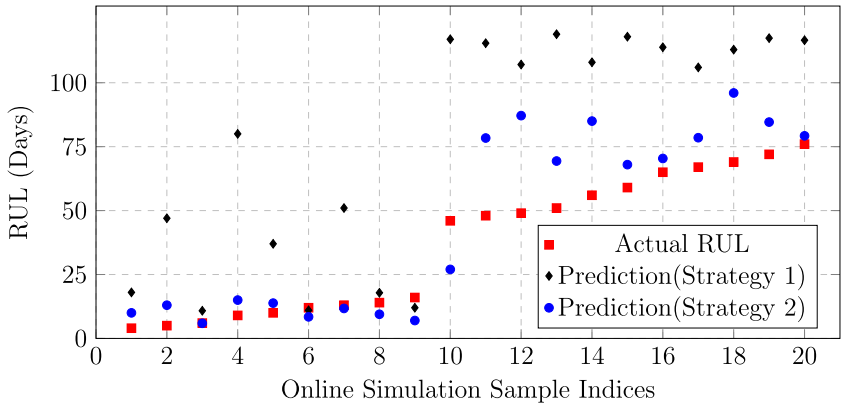

Fig. 14. Comparison of prediction strategy 1 and 2 in RUL estimation of the devices under simulation set.

4.2. Temporal analysis results

Fig. 14 depicts the actual and predicted RUL for devices under simulation set showing the effectiveness of Normalization Strategy 2 over Normalization Strategy 1 (Section 3.2.2). Normalization strategy 2 is also good for online analysis because we only use the 75 percentile of the observed values for normalization. We also see from Fig. 14 that the uncertainty of prediction is lower in case of devices having lower RUL and the prediction error is higher for devices that have enough time before they fail.

Fig. 15 shows precision, recall and F1 score of prediction of RUL. We have divided the results in two classes. Class 1 indicates the condition that the RUL is less than or equal to 10 days. Class 2 indicates that the RUL is greater than 10 days. If the actual and predicted RUL both fall in class 1, i.e., if the actual RUL is less than 10 and the predicted RUL is also less than 10, we consider it as true positive. If both the actual and predicted RUL are in class 2, then we consider it as true negative. If the actual RUL is in class 1 and the predicted RUL is in class 2, we consider it as false negative. If the actual RUL is in class 2 and the predicted RUL is in class 1, we consider it as false positive. In order to get a time series variation of these measures the process is repeated for seven consecutive days. An average Precision of 0.84, Recall of 0.72 and F1 score of 0.77 is achieved as evidenced by the plot. The flat nature of the curves indicate the consistency and robustness in decision making capabilities using the proposed approach over several consecutive days.

4.2.1. Uncertainty bounds

To establish the uncertainty bounds we study the results using the normalization strategy 2. We can note from Fig. 14 that the maximum error of prediction is 9 days. If the uncertainty interval is reduced to 6 days, 77% of the failure cases can be identified where the actual failure will lie in between the uncertainty interval of the predicted day of failure. As spatial analysis is relatively computationally inexpensive to perform, we chose the uncertainty interval as 9 days and perform the spatial analysis for all the days within that uncertainty interval.

4.3. Spatio-temporal analysis results

To validate our proposed approach for spatio-temporal analysis, we work with the same set of disks for which the temporal analysis results are shown in Fig. 14. Fig. 16 shows the actual day of failure and the predicted day of failure as

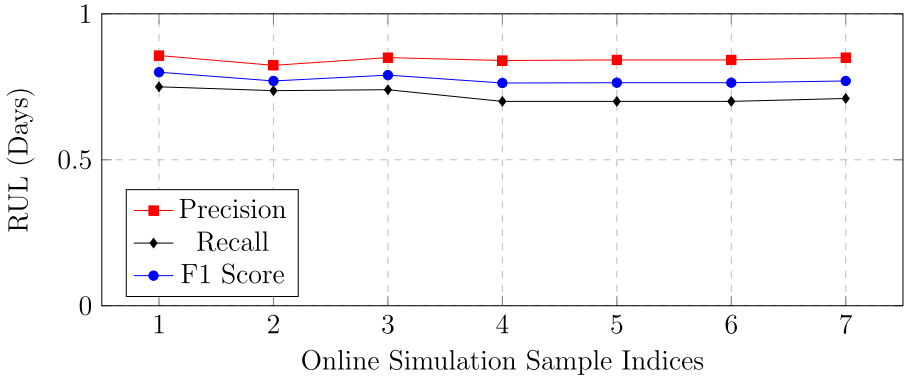

Fig. 15. Precision, recall and F1 score in predicting whether a device is going to fail within next ten days experimented over a period of seven consecutive days.

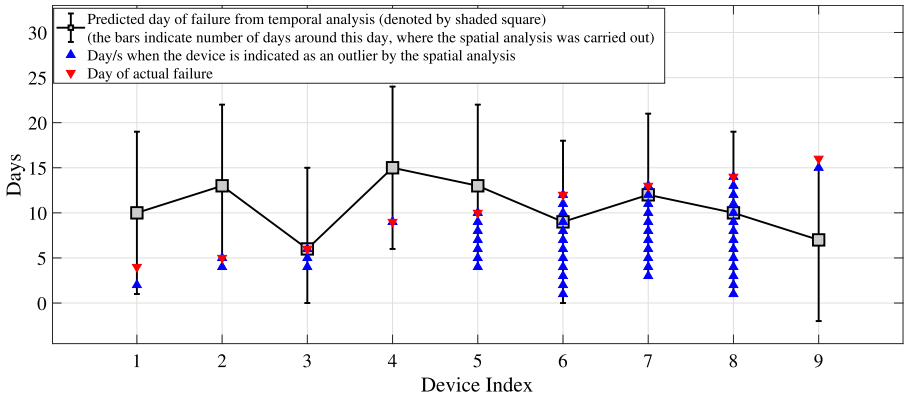

Fig. 16. Number of disk failures identified by the first and second step of spatial analysis compared to the total number of actual failures on each day.

per the temporal analysis for each device. The bars at each device index indicates the number of days before and after the predicted day of failure, when the spatial analysis has been carried out. If the predicted RUL is less than the lowest bound of the uncertainty interval, we start the spatial analysis for the device from the current day. It can be noted that some of the devices are identified as an outlier several days before and are continuously identified as anomalous up to the day of actual failure. This is because some of the SMART features correspond to cumulative error counts and once they cross a particular threshold they do not decrease till the day of failure and hence they appear in the outlier set for several consecutive days. On the other hand, Some devices such as device index 1 and 9 are indicated as outliers just one or two days before the failure and not on the actual day. This is because the normalized raw read error rate (SMART 1) was higher than the threshold for days before the failure – but not on the actual day of failure – this is one of the problems of spatial analysis. However, because the device was identified a few days before by spatial analysis and was also flagged by temporal analysis the data center operator can be more confident in taking decision about replacing it.

4.4. Improvement through spatio-temporal analysis

Since the precision of spatial analysis is too low, we cannot rely on only spatial analysis solely to identify disk failures. The spatial analysis identifies too many devices that are in the outlier set, out of which very few actually fail. We apply spatial analysis within the uncertainty interval of temporal analysis as identified in the combined inference analysis workflow shown in Fig. 1. So if a device is identified to fail by temporal analysis, the improvement by spatiotemporal analysis is based on how precisely we can identify the day of failure. For the devices that are going to fail in next 10 days, i.e., the actual RUL is less than or equal to 10 days, the mean absolute error (MAE) between the predicted day of failure and the actual day of failure is 5 days according to the temporal analysis. On the other hand, for the same set of devices the MAE between the actual day of failure and the day when the spatio-temporal analysis first identified the disk to fail was reduced to 2.4 days.

To estimate the precision and recall, we performed our analysis for five consecutive days where we calculated the total number of failures for model 'ST4000DM000' for each day and how many of them were identifiable by the temporal

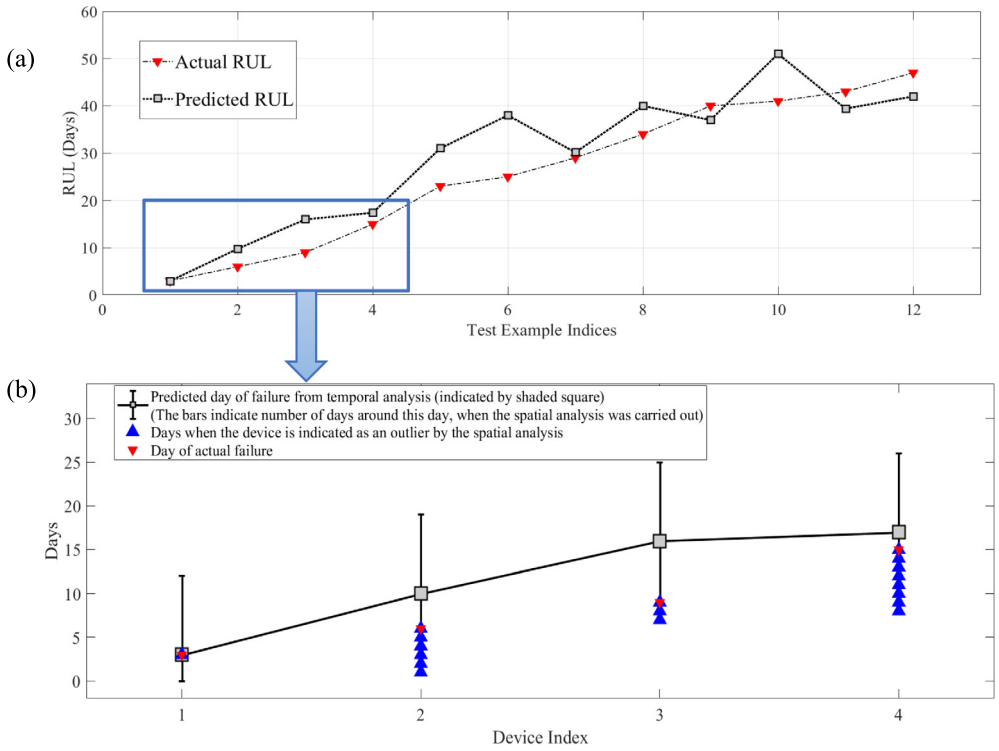

Fig. 17. (a) The figure shows the RUL prediction results using temporal analysis on a completely different model of Seagate (ST8000DM002), (b) Spatiotemporal analysis on the subset of devices where the predicted RUL is less than 20 days.

analysis within ten days of the actual day of failure. As spatio-temporal analysis is applied around the uncertainty interval of the temporally predicted day of failure, we identified how many of the disk failures identified by the temporal analysis was also identified by the spatio-temporal analysis that actually had the failure. We consider them as true positives. Now the spatial analysis can identify many false positives, but due to the reduction of analysis set due to the intersection with results from the temporal analysis approach, we can remove those false positives in the spatiotemporal analysis. Thus the number of false positives for the spatiotemporal analysis turns to zero in spite of having a large number of false positives in the spatial analysis which significantly improves the corresponding precision. The number of disks that were not identified by the spatiotemporal analysis but actually failed were considered as false negatives and was used in the recall calculation. Based on our analysis for five consecutive days, we got an average precision of 1 and an average recall of 0.664 for the spatio-temporal analysis.

4.5. Generalizability of the proposed approach

One of the major advantage of our proposed approach on spatio-temporal analysis of device health is that the mechanism is generalizable to other device models from the same manufacturer. For the temporal analysis, the neural network architecture was trained on Seagate model ST4000DM000. We applied the pre-trained model on ST4000DM000 to predict RUL of devices from another Seagate model ST8000DM002. Fig. 17(a) shows the results for RUL prediction on various devices using the temporal analysis. We applied the spatio-temporal analysis on the subset of these devices whose predicted RUL is less than twenty days. Fig. 17(b) shows the result for the spatio-temporal analysis where we see that the devices fall into the outlier category on the day/s close to the actual failure.

Fig. 1 shows a view of the overall workflow that should be exercised to implement the proposed spatiotemporal approach on any set of HDDs from a different manufacturer. Though the overall steps remain same, the feature selection for both spatial and temporal analysis, training and simulation data preprocessing have to be done again as different manufacturers report different sets of SMART indices. Also the hard disk failures from other manufacturers may be triggered from different thresholds on their corresponding feature values. Fig. 1 works as a general guideline of the high-level steps to be followed when repeating the procedure for devices from any manufacturer.

5. Conclusion and future work

This paper uses both spatial correlation and temporal progression characteristics of the health statistics of the devices to identify anomalous devices close to failure. The temporal analysis is based on a data-driven framework using deep LSTM

architectures for estimation of the remaining useful life of devices where the feature values corresponding to failure are not uniform across devices. The architecture proposed is efficient in predicting the remaining useful lives of devices having impending failures as well as segregating a set of devices on each day that show largely different feature patterns from the rest of the group. Although the proposed approaches are tested on the hard disk data, the combined spatio-temporal normalization, classification, and inference mechanisms are applicable to any generic time-series data capturing degrading system information. In the future, we expect to use this generalized spatio-temporal data analysis framework in various related applications aimed at real-time decision support.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is funded in part by the National Science Foundation under the award numbers 1840052 and 1818901 and an award from Cisco Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF and Cisco.

References

[1] L.A. Barroso, U. Hölzle, The datacenter as a computer: An introduction to the design of warehouse-scale machines, in: The Datacenter As a Computer: An Introduction To the Design of Warehouse-Scale Machines, 2008.
[2] B. Schroeder, G.A. Gibson, Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? in: Proceedings of the 5th USENIX Conference on File and Storage Technologies, FAST '07, USENIX Association, Berkeley, CA, USA, 2007, URL http://dl.acm.org/citation.cfm?id=1267903.1267904.
[3] G. Wang, L. Zhang, W. Xu, What can we learn from four years of data center hardware failures? in: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN, 2017, pp. 25–36, http://dx.doi.org/10.1109/DSN.2017.26.
[4] S. Basak, S. Sengupta, A. Dubey, Mechanisms for integrated feature normalization and remaining useful life estimation using LSTMs applied to hard-disks, in: 2019 IEEE International Conference on Smart Computing, SMARTCOMP, 2019, pp. 208–216, http://dx.doi.org/10.1109/SMARTCOMP.2019.00055.
[5] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, Long short term memory networks for anomaly detection in time series, in: Proceedings, Vol. 89, Presses universitaires de Louvain, 2015, pp. 89–94.
[6] H.D. Data, Stats/backblaze, 2019, URL: https://www.backblaze.com/b2/hard-drive-test-data.html.Checked24.03.
[7] N. Aussel, S. Jaulin, G. Gandon, Y. Petetin, E. Fazli, S. Chabridon, Predictive models of hard drive failures based on operational data, in: 2017 16th IEEE International Conference on Machine Learning and Applications, ICMLA, 2017, pp. 619–625, http://dx.doi.org/10.1109/ICMLA.2017.00-92.
[8] Y. Hu, S. Liu, H. Lu, H. Zhang, Remaining useful life model and assessment of mechanical products: A brief review and a note on the state space model method, Chin. J. Mech. Eng. 32 (1) (2019) 15, http://dx.doi.org/10.1186/s10033-019-0317-y.
[9] B. Schroeder, G.A. Gibson, Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? in: Proceedings of the 5th USENIX Conference on File and Storage Technologies, FAST '07, USENIX Association, USA, 2007, pp. 1–es.
[10] N. Gugulothu, T. Vishnu, P. Malhotra, L. Vig, P. Agarwal, G. Shroff, Predicting remaining useful life using time series embeddings based on recurrent neural networks, 2017, CoRR abs/1709.01073.
[11] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, LSTM-based encoder-decoder for multi-sensor anomaly detection, 2016, arXiv:1607.00148, CoRR abs/1607.00148 URL http://arxiv.org/abs/1607.00148.
[12] M.M. Botezatu, I. Giurgiu, J. Bogojeska, D. Wiesmann, Predicting disk replacement towards reliable data centers, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 39–48, http://dx.doi.org/10.1145/2939672.2939699.
[13] Y. Li, K. Liu, A.M. Foley, A. Zülke, M. Berecibar, E. Nanini-Maury, J.V. Mierlo, H.E. Hoster, Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review, Renew. Sustain. Energy Rev. 113 (2019) 109254, http://dx.doi.org/10.1016/j.rser.2019.109254, URL http://www.sciencedirect.com/science/article/pii/S136403211930454X.
[14] O.F. Eker, F. Camci, I.K. Jennions, A similarity-based prognostics approach for remaining useful life prediction, 2014.
[15] G.S. Babu, P. Zhao, X. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, in: DASFAA, 2016.
[16] F. Heimes, Recurrent neural networks for remaining useful life estimation, in: 2008 International Conference on Prognostics and Health Management, 2008, pp. 1–6, http://dx.doi.org/10.1109/PHM.2008.4711422.
[17] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, LSTM-based encoder-decoder for multi-sensor anomaly detection, 2016, CoRR abs/1607.00148.
[18] M.M. Botezatu, I. Giurgiu, J. Bogojeska, D. Wiesmann, Predicting disk replacement towards reliable data centers, in: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, ACM, New York, NY, USA, 2016, pp. 39–48, http://dx.doi.org/10.1145/2939672.2939699, URL http://doi.acm.org/10.1145/2939672.2939699.
[19] S. Sengupta, S. Basak, R.A. Peters, Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives, Mach. Learn. Knowl. Extr. 1 (1) (2018) 157–191, http://dx.doi.org/10.3390/make1010010, URL https://www.mdpi.com/2504-4990/1/1/10.
[20] S. Sengupta, S. Basak, R.A. Peters, Data clustering using a hybrid of fuzzy c-means and quantum-behaved particle swarm optimization, in: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC, 2018, pp. 137–142, http://dx.doi.org/10.1109/CCWC.2018.8301693.
[21] S. Sengupta, S. Basak, R.A. Peters, QDDS: A novel quantum swarm algorithm inspired by a double Dirac delta potential, in: 2018 IEEE Symposium Series on Computational Intelligence, SSCI, 2018, pp. 704–711, http://dx.doi.org/10.1109/SSCI.2018.8628792.
[22] J. Yu, B. Mo, D. Tang, H. Liu, J. Wan, Remaining useful life prediction for lithium-ion batteries using a quantum particle swarm optimization-based particle filter, 2017, http://dx.doi.org/10.6084/m9.figshare.4964996.v1.

[23] S. Basak, A. Dubey, B.P. Leao, Analyzing the cascading effect of traffic congestion using LSTM networks, in: Proceedings of IEEE Big Data, Los Angeles, CA, 2019.
[24] R. Sheikhpour, M.A. Sarram, S. Gharaghani, M.A.Z. Chahooki, A survey on semi-supervised feature selection methods, Pattern Recognit. 64 (2017) 141–158, http://dx.doi.org/10.1016/j.patcog.2016.11.003.
[25] Hard drive smart stats, 2020, (Accessed on July 2020), URL https://www.backblaze.com/blog/hard-drive-smart-stats/.
[26] What-smart-stats-indicate-hard-drive-failures, 2020, (Accessed on July 2020), URL https://www.backblaze.com/blog/what-smart-stats-indicate-hard-drive-failures/.
[27] S. Patel, S. Sihmar, A. Jatain, A study of hierarchical clustering algorithms, in: 2015 2nd International Conference on Computing for Sustainable Global Development, INDIACom, 2015, pp. 537–541.