On the Distribution Modeling of Heavy-Tailed Disk Failure Lifetime in Big Data Centers

Suayb S. Arslan, *Member, IEEE,* and Engin Zeydan, *Senior Member, IEEE*

***Abstract*—It has become commonplace to observe frequent multiple disk failures in big data centers in which thousands of drives operate simultaneously. Disks are typically protected by replication or erasure coding to guarantee a predetermined reliability. However, in order to optimize data protection, real life disk failure trends need to be modeled appropriately. The classical approach to modeling is to estimate the probability density function of failures using non-parametric estimation techniques such as Kernel Density Estimation (KDE). However, these techniques are suboptimal in the absence of the true underlying density function. Moreover, insufficient data may lead to overfitting. In this study, we propose to use a set of transformations to the collected failure data for almost perfect regression in the transform domain. Then, by inverse transformation, we analytically estimated the failure density through the efficient computation of moment generating functions and hence the density functions. Moreover, we developed a visualization platform to extract useful statistical information such as model-based mean time to failure. Our results indicate that for other heavy–tailed data, complex Gaussian Hypergeometric Distribution (GHD) and classical KDE approach can perform best if overfitting problem can be avoided and complexity burden is overtaken. On the other hand, we show that the failure distribution exhibits less complex Argus-like distribution after performing Box-Cox transformation up to appropriate scaling and shifting operations.**

***Index Terms*—Hard Disk Systems, Data Storage, Kernel Density Estimation, Modeling, Data Analytics.**

I. Introduction

HARD drives and more recent Solid State Drives (SSDs) have become the core/most common data storage units of today's data centers. These systems, that operate in close proximity and share the same geographical area, are affected by similar environmental factors, or the same hardware and network infrastructure, which increases the likelihood of these devices experiencing similar problems or undergoing close fault scenarios [1]. In the same way, the methods and technology used by manufacturers during the building process can create a faulty connection for storage devices. Therefore, a hardware or network problem can cause multiple storage devices to fail or become unavailable simultaneously in the network.

Since the reliability function $R(t)$ is closely related to cumulative distribution function $F(t)$ through the relationship $R(t) = 1 - F(t)$, it is of interest to estimate the probability density function (PDF) of failures to be able to quantify the reliability of storage devices. There are a number of challenges facing a reliability analyst to overcome in order to accurately predict the remaining lifetime of the storage devices in their data center. One of them is the hardness of the prediction model which has to take into account the correlated error scenarios [1]. This problem can be relieved by working on collected data where the correlation of failures would be captured by the data itself. On the other hand, due to the lack of knowledge about the underlying joint probability density function of failures, classical approaches to density estimation problem seems not to be directly applicable (smoothing parameter is usually missing).

Kernel Density Estimation (KDE) is one of the most commonly used technique for PDF estimation [2]. In that method, a kernel function along with a smoothing parameter are selected to fit a density function on the given data. In fact, the smoothing parameter is used to adjust the bias–variance trade–off in the literature. However, the optimal selection of the kernel function as well as the smoothing parameter depend heavily on the underlying PDF structure (light-tail, heavy-tail, etc). Since this PDF is usually unavailable, density estimation becomes a hard problem to solve. Yet, if there is a prior statistical knowledge about the data, then the selection of the kernel as well as the smoothing parameter can be made judiciously. For instance, if the underlying PDF is close to Gaussian, the kernel function being Gaussian (zero–mean and unit variance) and the smoothing parameter $\approx 1.06 \times \sigma N^{-0.2}$ are shown to perform best (in mean integrated square error sense) in the density estimation process, where σ is the sample standard deviation and N is the number of data samples [3].

One of the other methods to tackle the density estimation problem is to apply power transformations to the data so that the transformed data follows closely one of the well known distributions [4]. In other words, instead of solving the problem using non-parametric estimation techniques, we propose to solve it with parametric estimation in the transformed domain. In our study, we power transform the data using the Box-Cox transformation. The aim of the Box-Cox transformations is to ensure the usual assumptions for the linear model to hold that is the conventional assumption about the input distribution to be Guassian (Normal) distributed. Clearly not all data could be power-transformed to Normal and in our study, we shall demonstrate that the disk failure lifetime cannot be power-transformed to Normal either. However, we have realized that the transformed data (with slight scaling and translation) follows closely the Argus distribution. Furthermore, we propose a method to accurately estimate the underlying PDF as well as the related statistical properties efficiently. We have utilized Backblaze's publicly available dataset to confirm our

S. Arslan is with Department of Computer Engineering, MEF University, Istanbul, Turkey 34912. E-mail: arslans@mef.edu.tr..

E. Zeydan is with Centre Technologic de Telecomunicacions de Catalunya (CTTC), Castelldefels, Bercelona, Spain, 08860.

technique and quantify our results. It is important to note that our proposed technique is not restricted to disk data. Rather, any storage device failure data (particularly SSDs) whose underlying power-transformed PDF follows an Argus-like distribution can efficiently be estimated using the proposed method.

A. Related Works

With novel firmware support, hardware monitoring is now possible via low-level smart software modules that inspect modern storage disk devices and generates health data, collectively called SMART [5]. SMART data includes how long the device has been operational, the health status of the running hardware modules, operational data such as internal code errors or working temperature, etc. Early works of [1], [6] have demonstrated one of the pioneering analysis of hard disk failures over the metrics of Mean Time To Failure (MTTF) and field replacement rates of drives. As complementary to this approach, machine learning and statistical approaches for predictive modeling have surfaced in later analysis and development stages [7]–[13]. The authors in [7], [14] have used Bayesian network for Hard Disk Drives (HDD) failure predictions using SMART attributes. The papers in [9], [10] study remaining useful lifetime of HDD as a regression problem. The authors in [15] have proposed a cost sensitive ranking based machine learning method to select top-k faulty disk drives and minimize disk failure prediction errors. Online random forest based model in [13] and transfer learning modeling approaches [14] are proposed for disk failure prediction based on SMART attributes.

Many past studies have shown that SMART data can be effective in determining the life span of the storage devices and hence predict likely failures in the future. On the other hand, SMART data may not be readily available at all times or accurate enough for prediction use. In addition, in previous works such as [16] and [17], the importance of predicting major damage is emphasized and few techniques for predicting the failures have been developed without using SMART data. Once such a failure is predicted accurately, the data can readily be moved to a different storage device which saves the data from loss and necessitates no data repair action. This also reduces the extra bandwidth consumption due to background jobs such as rebuilding the lost data. To be able to eliminate false alarms and hence redundant data migrations, the accuracy of this operation is of great importance from an overall data durability perspective. However in the literature, there are still inefficiencies in many popular prediction models of real-life. These deficiencies result in increased amount of data storage due to the utilization of replication and/or erasure codes [18] and associated Markov models for reliability prediction [19]. However, data protection boosting methods through excessive redundancy (especially for replication) may lead to inefficient use of storage resources. As a result, data centers would need more storage devices which would result in more energy consumption and thus generate elevated heat cycles. The heat consumption level highly depends on the operating continuity of these devices and duty cycles [20]. Increased heat consumption not only shortens the lifetime of storage devices but also increases the power consumption of the overall system. Therefore, a right prediction model based on accurate failure distributions is highly needed to address these challenges of today's data storage community in which the accuracy has to be achieved based on the past failure trends.

Many researchers and industrial partners have also focused their efforts on analyzing real-world datasets. For example, there are plenty of works in the literature that concentrated on using Backblaze's public dataset to extract some useful models [21]–[28]. Other works such as [29] base their analysis (latent sector error analysis for reliability in this case) on proprietary data collected from production-ready storage systems. Backblaze's dataset includes all these such effects since it is collected from a real datacenter. However, such errors are only sensed if they are accumulated to lead to drive failures. Most of those studies are working on developing a predictive model for hard drive failures using machine learning and statistical approaches to improve storage reliability. A case study for predicting hard disk's time-to-failure using regression analysis is given by [24]. The authors in [21] have used various machine learning classification methods to predict hard drive failures. On the other hand, [22] studies statistical techniques to predict disk replacement with 98% accuracy using $30,000$ drives of Backblaze's dataset. However, prediction accuracy is closely tied with the accuracy of the PDF estimation of failures[1] and hence the outcome of this study can be used to complement these past literature works.

There are different metrics in the literature that are used to measure the data storage system reliability. For example, the authors in [30] propose NOrmalized Magnitude of Data Loss ($NOMDL_t$) metric (used to measure the expected amount of data lost per usable terabyte within mission time t) as a better alternative to standard Mean Time To Data Loss (MTTDL) metric [31], [32] or a novel metric named NOrmalized Magnitude of Data Unavailability (NOMDU) is proposed in [33] to measure the availability of data storage systems. The authors in [34] have used annualized failure rate (AFR) standard metric to analyze SMART data for describing a disk's fail-stop rate. We use MTTF/MTTDL metric to quantify the average reliability of a given protected array of storage devices. Although there are studies arguing that MTTF/MTTDL is a deficient tool for the absolute measurements and a system designer may be interested in the probability of failure for the first few years instead of MTTF/MTTDL. However, we argue that MTTF/MTTDL is still one of the most widely used intuitive reliability estimation metric that helps system designers to make right decisions beforehand. Moreover, this notorious reliability metric, based on exponential failure and repair times, has been shown in [35] to be insensitive to the actual distribution of failure/repair times as long as the constituent storage devices have much larger mean failure time than the mean repair time and operate independent of each other. Thus,

[1] Knowing the PDF implies that optimal detection/classifications can be performed over the data set. For instance given the true PDF, Bayesian classifier can perform MAP decoding and ensures optimal detection. Thus, most of the difficulty lies in the estimation of potentially multi-variate PDF that characterizes the system dynamics. However, the accuracy of the solution is closely tied with the volume of the dataset and its peculiar feature richness. Otherwise, other methods are preferred for classification and prediction.

using MTTF/MTTDL and known distributions, we can simply generate an answer to the probability of failure for the first few years pretty accurately. Plus, closed-form expressions for MTTDL are shown to be possible in [19] for more complicated general cases and such analytic expressions usually help our intuition for modeling error-tolerant data storage systems.

To determine the distribution of failure lifetime, there also exists various approaches for fitting the lifetime distribution of failures into an existing distribution in the literature [27]. A Generalized Limited Failure Population (GLFP) model using a hierarchical modeling approach for the estimation of lifetime distribution is studied in [27]. Enhancing reliability of hard disk drives has also been the focus of research efforts in recent decades [27], [36]. The authors in [36] develop an analytic model that uses both contextual information and storage utilization to increase overall storage reliability. However, none of them utilizes a comparative approach in terms of fitted distribution complexity evaluation by resorting to data transformation methods by analyzing the widely available recent BackBlaze dataset. Many reliability works lead by Elerath et al. concentrate on modeling disk failures with a Weibull distribution regarding the field data obtained from tens of thousands of enterprise-class disks [31], [37]–[46]. Different than those approaches, our work in this paper has evaluated how other known distributions work with the collected field dataset in comparison with Weibull distribution (e.g. Gaussian Hypergeometric Distribution (GHD) before Box-Cox transformation and Argus distribution after Box-Cox transformation).

In this study, although the existing dataset can well fit into a highly complex distributions such as GHD, we show that Box-Cox transformed disk failure life time can resemble to a less complex distribution such as Argus-like distribution up to an appropriate scaling and shifting. In the past, Box-Cox transformation has been observed to be useful for applications in Big Data [47], [48]. In the literature, known distributions with larger parameter space has been applied to different field data and experiments. For instance, the authors in [49] have used Gaussian hypergeometric function to fit real data with non-homogeneous profiles including data from food stores in Ljubljana, insurance claims and number of accidents. However, the pitfall associated with this approach is the outstanding complexity of accurately estimating many parameters of the system. In fact, transformation of data and using a known distribution with less parameter space might be usually preferable from the complexity point of view.

B. Our Contributions

There are several notable distinctions between the above outlined papers and our contributions. In this paper, just like several past research, we have utilized BackBlaze's open source dataset [50] to build our performance evaluation platform for efficiently modeling the true behaviour of hard disk failure lifetime. This platform is useful not only for loading data and obtain quick analytics but also for efficient indexing, finding, and determining required distributions for the next modeling phase. Our contributions in this paper can be summarized as follows:

- For the first time in literature, we have shown that hard disk failure lifetime distribution of the observed dataset closely follows GHD and that Box-Cox transformed data closely follows an Argus-like distribution up to a proper scaling and shifting where the transformation-Argus combination is observed to be computationally more efficient than the former.
- Contrary to classical approaches such as KDE, we derived novel and alternative methods to find known statistical metrics about the lifetime and hence the reliability such as MTTF (in closed form) for non-repairable systems as well as the density estimation of the transformed lifetime data through moment generating functions (MGFs). Our proposed scheme illustrates a practical data utilization strategy and analytically derive closed form expressions to estimate the reliability which in our opinion would be quite valuable for reliability analysts as well as engineers to quickly get a handle on their current system.
- We have also considered/compared other distributions such as exponential (single parameter estimation), Weibull (two parameters estimation) as well as GHD (four parameters estimation) functions to compare with the performances of our proposed Argus-based MTTF estimations to extract the trade-off between complexity and accuracy of distribution fitting on to the dataset (similar to the well known high bias and variance trade-off). We have observed that for heavy-tailed data, as the number of estimated parameters increase, the chances of overfitting gets lot higher. Hence, the proposed transformation for Argus-like distribution estimation achieves a good trade-off point.

Finally, thanks to the built-in data visualization module and analysis platform, we manipulate and illustrate the hard disk data where key findings can be reported using a user-friendly Kibana interface. We have also detailed which model of the hard disk manufacturer ~~can~~ performs better than the others.

C. Notation and Organization

Throughout the document, $Pr\{.\}$ denotes the probability, $\mathbb{E}[.]$ is the expectation operator, ${}_{\boldsymbol{2}}\boldsymbol{F}_{\boldsymbol{1}}$ is Hypergeometric function, $I_1(.)$ is the Modified Bessel function of the first kind of order one, $M_Y(.)$ is the moment generating function of Y, $f_X(.)$ denotes the PDF whereas $F_X(.)$ is the corresponding Cumulative Distribution Function (CDF) of the random variable X, $BC_\theta(.)$ denotes the Box-Cox transformation, $sup\{.\}$ and $inf\{.\}$ denotes the infimum and supremum. The sets are denoted by upper case calligraphic symbols. The scalars are represented by regular symbols and vectors are denoted by bold face regular letters, e.g., $\boldsymbol{x}$ where $\boldsymbol{x}(k)$ denotes the k-th element of $\boldsymbol{x}$.

The rest of the paper is organized as follows: In Section II, we introduce some of the basic definitions and review few applicable distributions for modeling the disk failure lifetime. Additionally, Box-Cox transformation is generalized and associated mean and moment generating functions are derived. In Section III, we provide the details of data preprocessing, estimation of statistical information and a novel low-cost iterative mean time to failure procedure. In Section IV, we provide the details and properties of the hard disk data evaluation platform developed specifically for our study. Section V outlines some of the numerical results using Backblaze's data. Finally in Section VI, we present conclusions and some of the potential future directions.

II. Disk Failure Lifetime Distribution

In our system, the set of multiple hard disk manufacturers is represented by the set $\mathcal{D} = \{m_1, m_2, , \dots, m_D\}$ with D different manufacturers. Each hard disk manufacturer m_i with $1 \leq i \leq D$ has a set of serial numbers given by the set $\mathcal{S}_i = \{s_1, s_2, \dots, s_{S_i}\}$ with S_i different serial numbers for a given manufacturer m_i and the set of lifetime for each serial number in $\mathcal{S}_i$ is represented by the set $\mathcal{Y}_i = \{o_1, o_2, , \dots, o_{S_i}\}$. In general, the disk failure lifetime is defined as follows,

$$y_j = T_e^{s_j} - T_b^{s_j} \quad \text{with} \quad 1 \leq j \leq S_i \tag{1}$$

where $T_b^{s_j}$ and $T_e^{s_j}$ represent the disk's production time (beginning of operation) and failure time (end of operation) stamps, respectively. In this paper, y_j will be expressed in unit of days.

Definition 2.1 (Heavy Tailed (HT) distribution [51]). *For a random variable X if*

$$\lim_{x \mapsto \infty} \sup e^{\lambda x}(1 - F_X(x)) = \infty, \;\; \forall \lambda > 0, \tag{2}$$

where $F_X(x) = Pr(X \leq x)$ denotes the cumulative distribution of random variable X, then random variable X is HT distributed.

In other words, for HT distributions $E[e^{\lambda X}] = \infty, \;\; \forall \lambda > 0$. An HT distributed random variable (e.g, Pareto, Cauchy, log-normal and Weibull) has a slower tail variance decay compared to that of exponential distribution which explains the name "heavy tail". Let us start by defining few important distributions for our study from the theory of statistics in the next subsections.

A. Gaussian Hypergeometric Distribution (GHD)

The probability density function for a dummy random variable A which has the four-parameters $(\alpha, \beta, \gamma, z)$ for GHD is given as,

$$f_A(x; \alpha, \beta, \gamma, z) = Cx^{\alpha-1}(1-x)^{\beta-1}(1+zx)^{-\gamma}, \tag{3}$$

where $0 \leq x \leq 1$, $\alpha > 0$, $\beta > 0$ and $C = B(\alpha, \beta)^{-1} {}_{\mathbf{2}}\boldsymbol{F}_{\mathbf{1}}(\gamma, \alpha; \alpha + \beta; -z)^{-1}$ and ${}_{\mathbf{2}}\boldsymbol{F}_{\mathbf{1}}$ is the hypergeometric function defined as

$${}_{\mathbf{2}}\boldsymbol{F}_{\mathbf{1}}(\alpha, \beta, \gamma, z) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!}. \tag{4}$$

and $()_n$ is the Pochhammer symbol [52].

Note that many discrete probability distributions such as Poisson, Binomial have probability generating functions which can be expressed as a special case of the Gaussian hypergeometric function as outlined in [49].

B. Argus Distribution

The probability density function for a dummy random variable A which has two-parameters (χ, c) for Argus distribution, is given for $0 \leq x < c$,

$$f_A(x; \chi, c) = \frac{\chi^3}{\sqrt{2\pi}\Psi(\chi)} \frac{x}{c^2} \sqrt{1 - \frac{x^2}{c^2}} \exp\left\{-\frac{1}{2}\chi^2\left(1 - \frac{x^2}{c^2}\right)\right\}, \tag{5}$$

where $\Psi(\chi) = \Phi(\chi) - \chi\phi(\chi) - 1/2$. Moreover, $\Phi(\chi)$ and $\phi(\chi)$ denote the cumulative probability and density functions for the normalized Gaussian (standard Normal) distribution, respectively. In that regard, as $c, x \to \infty$, we can easily show that Argus has HT distribution by observing,

$$\lim_{x, c \mapsto \infty} \frac{1 - F_A(x)}{e^{-\lambda x}} = \lim_{\substack{x \mapsto c \\ c \mapsto \infty}} \frac{\Psi(\chi\sqrt{1 - x^2/c^2})}{\Psi(\chi)e^{-\lambda x}} \tag{6}$$

$$\geq \lim_{c \mapsto \infty} \frac{\Psi(0)e^{\lambda c}}{\Psi(\chi)} = \infty. \tag{7}$$

where $F_A(x) = 1 - \Psi(\chi\sqrt{1 - x^2/c^2})/\Psi(\chi)$. In recent years, it has been observed that most of the data collected from many real world scenarios exhibit statistical behaviours that can be best described by HT distributions. These examples include file size distributions [53], connection times [54], and web page sizes of the Internet [53]. In this paper, we will show that the hard disk failure lifetime distribution also follows a HT distribution behavior using the real-world data.

C. Weibull Distribution

Previous works conducted on HDD lifetime have demonstrated that the disk failure may be better specified by Weibull distribution using the field data of over 120,000 HDDs that operated for up to 6,000 hours [40]. A two-parameter Weibull probability density function, with shape parameter c_w and the scale parameter a, can be described as

$$f_A(x, c_w) = \frac{c_w}{a}\left(\frac{x}{a}\right)^{c_w - 1} exp\left\{-\left(\frac{x}{a}\right)^{c_w}\right\} \tag{8}$$

In a Weibull distribution, the shape parameter c_w indicates whether the HDD failure rate is decreasing (in case of $c_w <$ 1.0), constant (in case of $c_w = 1$), or increasing (in case of $c_w >$ 1.0) [31].

D. Exponential Distribution

For the sake of completeness, we also provide a popular single-parameter distribution i.e., exponential distribution. The probability density function for a dummy random variable A which has one-parameter λ for exponential distribution is given by

$$f_A(x, \lambda) = \lambda e^{-\lambda x} \tag{9}$$

which is typically used with Poisson distribution and models inter-arrival times of many real-time processes.

E. Power Transformations

In this study, we propose to apply a Box-Cox power transformation to the hard disk failure lifetime data, denoted by y_j, to demonstrate that the transformed disk failure lifetime may be distributed according to Argus distribution. More formally, let Y denote the lifetime distribution of a set of disk drives (realizations are represented by y_j), then we define the new random variable $X = BC_\theta(Y)$ which is the Box-Cox transformed version of Y where

$$BC_\theta(Y) = \begin{cases} (Y^\theta - 1)/\theta & \text{if } \theta \neq 0 \\ \log(Y) & \text{if } \theta = 0 \end{cases} \tag{10}$$

and θ is the parameter of the transformation. For inverse Box-Cox transformation, i.e., $Y = BC_\theta^{-1}(X)$, we use

$$BC_\theta^{-1}(X) = \begin{cases} (1 + \theta X)^{1/\theta} & \text{if } \theta \neq 0 \\ e^X & \text{if } \theta = 0 \end{cases} \tag{11}$$

One of the objectives of this study is to show that the distribution X can closely be characterized by Argus distribution using real-world data. Note that we have

$$\begin{aligned} Pr\{Y \leq y\} = Pr\{BC^{-1}(X) \leq y\} &= Pr\{X \leq BC(y)\} \\ &= F_A(BC(y)) \end{aligned} \tag{12}$$

Thus, the probability density function of Y shall be given by the derivative

$$\frac{d}{dy}F_A(BC_\theta(y)) = f_A(BC_\theta(y), \chi, c)\frac{d}{dy}BC_\theta(y) \tag{13}$$

which can be explicitly expressed as,

$$f_Y(y, \theta, \chi, c) = \begin{cases} f_A((y^\theta - 1)/\theta, \chi, c)y^{\theta - 1} & \text{if } \theta > 0 \\ f_A(\log(y), \chi, c)/y & \text{if } \theta = 0 \end{cases} \tag{14}$$

where $1 \leq y < \sqrt[\theta]{c\theta + 1}$ for $\theta > 0$ and $1 \leq y < e^c$ for $\theta = 0$. Note that it is not hard to show that $f_Y(y, \theta, \chi, c)$ is a valid distribution.

If we let y_j to be the data samples and x_j to be the transformed version of these data samples[2], the transformation parameter θ (optimal to be denoted by θ^*) is chosen to maximize the Box-Cox log-likelihood function [55] which is given by

$$\begin{aligned} \theta^* = \arg\max_\theta \sum_j \log(y_j)^{\theta - 1} \\ - \frac{N}{2}\ln\left(\frac{1}{N}\sum_j (x_j - \overline{x})^2\right) \end{aligned} \tag{15}$$

where N is the total number of data samples used in the optimization and $\overline{x} = \frac{1}{N}\sum_j x_j$ is the mean. For instance, if we only consider the disk manufacturer m_i, then $N = S_i$. We could alternatively consider all of the manufacturers in which case we shall have $N = \sum_i S_i$.

F. Mean Time To Failure (MTTF)

In this subsection, we will argue that MTTF can be calculated using the moments of Argus distribution paramaterized by c and χ. MTTF, being the most commonly used reliability metric, is defined to be the length of time a disk device is expected to last its healthy operation. Although MTTF is one of many ways to evaluate the reliability of hard disk drives, it is probably the most commonly known. Simply put, MTTF can be calculated as the arithmetic mean (average) time between the initial onset and the instant when the driver fails for the first time. The following theorem characterizes it explicitly.

Theorem 2.2. *For a given transformation parameter $\theta \geq 0$, the mean time to failure (the mean value of the lifetime distribution defined on x) is given by*

$$MTTF = \begin{cases} \sum_{i=0}^{\infty} \frac{\mu_A(i)}{i!} & \textit{if } \theta = 0 \\ \sum_{k=0}^{\infty} \binom{1/\theta}{k}\theta^k \mu_A(k) & \textit{if } \theta > 0 \end{cases} \tag{16}$$

where $\mu_A(i)$ represents the ith moment of the Argus distribution and the equality for $\theta > 0$ in (16) *converges for any real θ satisfying $-1/\theta < x \leq 1/\theta$.*

Proof: Using the definition, we have

$$\begin{aligned} \text{MTTF} = \mathbb{E}[Y] &= \int y f_Y(y, \theta, \chi, c)dy & (17) \\ &= \int BC_\theta^{-1}(x) f_A(x; \chi, c)dx. & (18) \end{aligned}$$

Since Box-Cox transformation is invertible, using Taylor series expansion, we can rewrite equation (18) in terms of an infinite sum for different values of the parameter θ. If $\theta = 0$ using equation (10), we can rewrite it as

$$\mathbb{E}[Y] = \int (1 + x + \frac{x^2}{2} + \dots) f_A(x; \chi, c)dx = \sum_{i=0}^{\infty} \frac{\mu_A(i)}{i!} \tag{19}$$

where we used distribution law and $\mu_A(i)$ represents the ith moment of the Argus distribution given by $\mu_A(i) \triangleq \int x^i f_A(x; \chi, c)dx$. The first and second moments can be expressed in closed form as follows,

$$\begin{aligned} \mu_A(1) &= c\sqrt{\frac{\pi}{8}}\frac{\chi e^{-\chi^2/4} I_1(\chi^2/4)}{\Psi(\chi)} & (20) \\ \mu_A(2) &= c^2\left(1 - \frac{3}{\chi^2} + \frac{\chi\phi(\chi)}{\Psi(\chi)}\right) & (21) \end{aligned}$$

where $I_1(.)$ is the Modified Bessel function of the first kind of order one. Similarly for $\theta > 0$, we can write

$$\begin{aligned} \mathbb{E}[Y] &= \int_0^c (1 + x\theta)^{1/\theta} f_A(x; \chi, c)dx & (22) \\ &= \int_0^c \sum_{k=0}^{\infty} \binom{1/\theta}{k} (x\theta)^k f_A(x; \chi, c)dx & (23) \\ &= \sum_{k=0}^{\infty} \binom{1/\theta}{k} \theta^k \mu_A(k) & (24) \end{aligned}$$

where the equality in (23) converges for any real θ satisfying $-1/\theta < x \leq 1/\theta$ and hence the condition. ■

If we make sure $|c\theta| \leq 1$, then we will guarantee that the infinite sum will converge and MTTF can be found using

[2] One can think of x_j and y_j as realizations of random variables X and Y, respectively

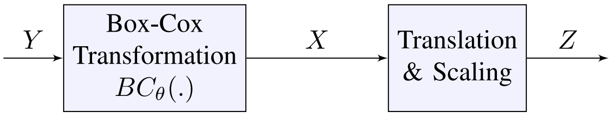


Fig. 1: Summary of generalized Box-Cox Transformation. Translation and Scaling is introduced to better adapt to data and improve accuracy.

Equation (24). In other words, enough number of terms in the sum shall yield a good estimate of the MTTF using the parameters of the Argus distribution which are extracted from the data through parameter estimation techniques. Note that since $1/\theta$ is real, we need to use the Gamma function to generalize the binomial coefficient for non-integer arguments by using the following equality,

$$\binom{1/\theta}{k} = \frac{\Gamma(1/\theta+1)}{\Gamma(k+1)\Gamma(1/\theta-k+1)} \tag{25}$$

where $\Gamma(z)$ is readily available in various software tools and computes the integral $\Gamma(t) = \int_0^\infty s^{t-1}e^{-s}ds$.

It is typical that the parameter c is assumed to be known, and χ is estimated from the samples x_j using the maximum likelihood approach. The estimator turns out to be a function of the sample second moment, and is given as a solution to the following non-linear equation

$$1 - \frac{3}{\chi^2} + \frac{\chi\phi(\chi)}{\Psi(\chi)} = \frac{1}{n}\sum_{j=1}^{n}\frac{x_j^2}{c^2} \tag{26}$$

which can simply be obtained by setting the second moment of the distribution to the sample second moment[3].

G. Generalization of Box-Cox Transformation

Due to real world translation and scaling effects, we need a generalization of the Box-Cox transformation that includes these effects to boost accuracy. As summarized in Fig. 1, we have two forms of transformations one of which is the basic Box-Cox transformation ($BC_\theta(.)$) which is followed by translation and scaling operations, collectively referred as $GBC_\theta(.)$. Accordingly, let us define a new random variable

$$\begin{aligned} Z := GBC_\theta(Y) &= (BC_\theta(Y) - loc)/scale && (27)\\ &= (X - loc)/scale && (28) \end{aligned}$$

where we shall denote loc as l and $scale$ as s from now on and GBC_θ stands for the generalized box-cox transformation. In this case, the distribution of the random variable Z is deemed to be closely characterized by Argus distribution.

Theorem 2.3. *For given transformation, translation and scaling parameters, i.e., θ, s, l, respectively, the mean of the lifetime distribution defined on x is given by*

$$MTTF = \begin{cases} e^l + \sum_{i=0}^{\infty}\frac{s^i\mu_A(i)}{i!} & \textit{if } \theta = 0\\ \sum_{k=0}^{\infty}\binom{1/\theta}{k}(l\theta)^k\sum_{i=0}^{k}\binom{k}{i}\left(\frac{s}{l}\right)^i\mu_A(i) & \textit{if } \theta > 0 \end{cases} \tag{29}$$

where $\mu_A(i)$ represents the ith moment of the Argus distribution and the equality for $\theta > 0$ in (29) *converges for any real θ satisfying $-1/\theta < sx + l \leq 1/\theta$.*

Proof: The probability density function of Y can be computed by taking the derivative of the cumulative distribution function of Z as given by

$$\begin{aligned} f_Y(y,\theta,\chi,c) &= \frac{d}{dy}F_A(GBC_\theta(y))\\ &= \frac{d}{dz}F_A(GBC_\theta(y))\frac{dz}{dy}\\ &= \frac{d}{dz}F_A(GBC_\theta(y))\frac{d}{dy}\left(\frac{BC_\theta(y)-l}{s}\right)\\ &= \frac{f_A(GBC_\theta(y),\chi,c)}{s}\frac{d}{dy}BC_\theta(y) \end{aligned}$$

and we can similarly express MTTF as

$$\begin{aligned} \mathbb{E}[Y] &= \int yf_Y(y,\theta,\chi,c)dy && (30)\\ &= \frac{1}{s}\int GBC_\theta^{-1}(z)f_A((x-l)/s;\chi,c)dx. && (31)\\ &= \frac{1}{s}\int BC_\theta^{-1}(x)f_A((x-l)/s;\chi,c)dx. && (32)\\ &= \int BC_\theta^{-1}(sz+l)f_A(z;\chi,c)dz. && (33) \end{aligned}$$

where if $s = 1$ and $l = 0$, this becomes identical to Equation (18) of Theorem 2.2. Using this equation, we can finally get

$$\mathbb{E}[Y] = \begin{cases} \int_0^c e^{sz+l}f_A(z;\chi,c)dz & \text{if } \theta = 0\\ \int_0^c(1+\theta(sz+l))^{1/\theta}f_A(z;\chi,c)dz & \text{if } \theta > 0 \end{cases} \tag{34}$$

We can further use the Taylor expansions as before to obtain sum-only expressions. Since different cases require different treatment, we will explore both cases separately.

1) Case $\theta = 0$: It can be shown to be of the form for $l > 0$

$$\begin{aligned} \mathbb{E}[Y] &= \int_0^c e^{sz+l}f_A(z;\chi,c)dz && (35)\\ &= e^l + \int_0^c\left(1 + sz + \frac{s^2z^2}{2} + \dots\right)f_A(z;\chi,c)dz\\ &= e^l + \sum_{i=0}^{\infty}\frac{s^i\mu_A(i)}{i!} && (36) \end{aligned}$$

[3] In most of the software tools, c is set to some constant and χ is estimated for the set of given data samples which reduces it to single parameter estimation. For instance c is set to unity in *scipy* python library without loss of generality.

2) Case $\theta > 0$: In this case, we have

$$\begin{aligned}
\mathbb{E}[Y] &= \int_0^c (1+\theta(sz+l))^{1/\theta} f_A(z;\chi,c) && (37)\\
&= \int_0^c \sum_{k=0}^{\infty} \binom{1/\theta}{k} (sz+l)^k \theta^k f_A(z;\chi,c)dz && (38)\\
&= \sum_{k=0}^{\infty} \binom{1/\theta}{k} \sum_{i=0}^{k} \binom{k}{i} s^i l^{k-i} \theta^k \mu_A(i) && (39)\\
&= \sum_{k=0}^{\infty} \binom{1/\theta}{k} (l\theta)^k \sum_{i=0}^{k} \binom{k}{i} \left(\frac{s}{l}\right)^i \mu_A(i) && (40)
\end{aligned}$$

■

Note that if $-1/\theta < sc + l \leq 1/\theta$ and together with Equation (25) (for $\sum_{k=0}^{\infty} \binom{1/\theta}{k}$), we can use (40) to calculate MTTF. Later in numerical results section, for all practical cases this inequality will be shown to be typically satisfied which makes Eqn. (40) applicable.

H. Higher order statistics and Moment generating function

It is of interest to estimate the higher order of moments of Y to calculate the moment generating function of the underlying probability distribution function of the data. Fortunately, it turns out that computation of higher moments are not very different than Theorem 2.3 and can be used to estimate the moment generating function. The following corollary characterizes it explicitly.

Corollary 2.4. *The moment generating function of the disk lifetime, denoted by $M_Y(t)$, is given by*

$$M_Y(t) = \sum_{n=0}^{\infty} \frac{t^n}{n!} \mathbb{E}[Y^n] \quad (41)$$

where

$$\mathbb{E}[Y^n] = \begin{cases} e^{nl} + \sum_{i=0}^{\infty} \frac{s^i n^i \mu_A(i)}{i!} & \text{if } \theta = 0 \\ \sum_{k=0}^{\infty} \binom{n/\theta}{k} (l\theta)^k \sum_{i=0}^{k} \binom{k}{i} \left(\frac{s}{l}\right)^i \mu_A(i) & \text{if } \theta > 0 \end{cases} \quad (42)$$

Proof: Similar to Theorem 2.3, for $\theta > 0$ and $-n/\theta < sc + l \leq n/\theta$, we have

$$\begin{aligned}
\mathbb{E}[Y^n] &= \int y^n f_Y(y,\theta,\chi,c)dy && (43)\\
&= \int_0^c (1+\theta(sz+l))^{n/\theta} f_A(z;\chi,c) && (44)\\
&= \int_0^c \sum_{k=0}^{\infty} \binom{n/\theta}{k} (sz+l)^k \theta^k f_A(z;\chi,c)dz && (45)\\
&= \sum_{k=0}^{\infty} \binom{n/\theta}{k} \sum_{i=0}^{k} \binom{k}{i} s^i l^{k-i} \theta^k \mu_A(i) && (46)\\
&= \sum_{k=0}^{\infty} \binom{n/\theta}{k} (l\theta)^k \sum_{i=0}^{k} \binom{k}{i} \left(\frac{s}{l}\right)^i \mu_A(i) && (47)
\end{aligned}$$

Note that if $|sc + l| \leq 1/\theta$, then the necessary condition $-n/\theta < sc + l \leq n/\theta$ is automatically satisfied for $n > 1$. Additionally, similar extension can be done for $\theta \to 0$ to finally obtain

$$\begin{aligned}
\mathbb{E}[Y^n] &= \lim_{\theta \to 0} \int_0^c (1+\theta(sz+l))^{n/\theta} f_A(z;\chi,c)\\
&= e^{nl} + \sum_{i=0}^{\infty} \frac{s^i n^i \mu_A(i)}{i!}
\end{aligned} \quad (48)$$

Finally, the moment generating function of Y, denoted as $M_Y(t)$, is defined to be of the form

$$M_Y(t) = \int e^{ty} f_Y(y,\theta,\chi,c)dy = \sum_{n=0}^{\infty} \frac{t^n}{n!} \mathbb{E}[Y^n] \quad (49)$$

■

We can compute the moments by simply taking derivatives of $M_Y(t)$. In the next section, we provide how to use the data in light of our findings in this section, after preprocessing, in order to estimate MTTF, MGF and CDF using a low complexity iterative method.

III. Utilization of Real Data

A. Support mapping and an iterative solution

One of the things we notice in the analysis of the previous section is that the support of Y needs to be $[1, \sqrt[\theta]{c\theta + 1})$ which stems from the fact that the random variable that is modelled to be distributed according to Argus distribution needs to have support $[0, c)$. In addition, without such scaling operation on the support, we would not be able to use equations (16) and (29) because the associated conditions would not be met.

Suppose that the random variable that represents the real data is represented by Y_d with a support not necessarily equal to $[1, \sqrt[\theta]{c\theta + 1})$. Also let $y_j^{(d)}$ to be the real data samples (collected field data) and x_j to be the transformed version of scaled samples y_j, introduced earlier as data samples. Since real data samples $y_j^{(d)}$ needs to be appropriately scaled to meet the condition imposed by the support of Y, it needs to go through a transformation. It is not hard to find that the following transformation refines the expected support for $\theta > 0$,

$$y_j = ((c\theta + 1)^{1/\theta} - 1)\overline{y}_j^{(d)} + 1 \quad (50)$$

where $\overline{y}_j^{(d)}$ is the scaled version of $y_j^{(d)}$ and is given by

$$\overline{y}_j^{(d)} = \frac{y_j^{(d)} - \inf\{Y_d\}}{\sup\{Y_d\} - \inf\{Y_d\}}. \quad (51)$$

and $\sup\{Y_d\}$ and $\inf\{Y_d\}$ represents the maximum and minimum of $y_j^{(d)}$, respectively. Note that if $\theta \to 0$, we have $(c\theta + 1)^{1/\theta} \to e^c$ which will not have any θ dependence. The transformation parameter θ (optimal to be denoted by θ^*) is chosen to maximize the Box-Cox log-likelihood function which is given by our previous optimization problem in equation (15), where $x_j = BC_\theta(y_j)$ is as defined earlier and N is the total number of data samples used in the optimization with $\overline{x} = \frac{1}{N}\sum_j x_j$.

However, the issue is that this one–shot optimization can be hard to solve. An alternative is to iteratively optimize θ to

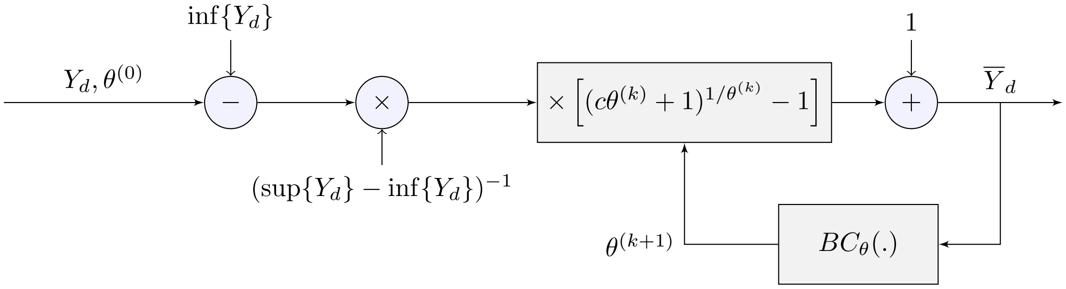

$\inf\{Y_d\}$
$Y_d, \theta^{(0)}$
$-$
$\times$
$(\sup\{Y_d\} - \inf\{Y_d\})^{-1}$
$\times \left[(c\theta^{(k)}+1)^{1/\theta^{(k)}} - 1\right]$
1
$+$
$\overline{Y}_d$
$\theta^{(k+1)}$
$BC_\theta(.)$

Fig. 2: Iterative optimization of θ using real data sets which loops until $\theta^{k+1} - \theta^k < \epsilon$ where ϵ is some small tolerance. Iterations are defined for $c\theta^{(k)} + 1 \geq 0$ otherwise, iterations cease.

obtain θ^* by using $BC_\theta(.)$ as a standard block as shown in Fig. 2. The initial value $\theta^{(1)}$ is obtained by setting $y_j^{(0)} = \overline{y}_j^{(d)}$ where for $k > 0$ we have

$$y_j^{(k)} = ((c\theta^{(k)} + 1)^{1/\theta^{(k)}} - 1)\overline{y}_j^{(d)} + 1 \tag{52}$$

where $\theta^{(k)}$ is the value of θ at the k-th iteration. Note here that the subsequent iterations are defined only for $\theta^{(k)} \geq -1/c$. The $(k+1)$-th value of θ is given by

$$\theta^{(k+1)} = \arg\max_\theta \sum_j \log(y_j^{(k)})^{\theta-1} - \frac{N}{2}\ln\sigma_y^{(k)} \tag{53}$$

where

$$\sigma_y^{(k)} = \frac{1}{N}\sum_j \left(BC_\theta(y_j^{(k)}) - \frac{1}{N}\sum_j BC_\theta(y_j^{(k)}) \right)^2 \tag{54}$$

Iterations continue until a preset threshold (ϵ) is exceeded i.e., we can find minimum $k^* \in \mathbb{N}$ such that $\theta^{(k^*+1)} - \theta^{(k^*)} < \epsilon$. Thus, the final support–mapped data is given by

$$y_j = y_j^{(k^*)} = y_j^{(k^*+1)} = ((c\theta^{(k^*)} + 1)^{1/\theta^{(k^*)}} - 1)\overline{y}_j^{(d)} + 1 \tag{55}$$

We note that convergence is established if $c\theta^{(k)} + 1 \geq 0$. Otherwise the limit does not exist i.e., $\limsup_{k\to\infty} \theta^{(k)} \neq \liminf_{k\to\infty} \theta^{(k)}$ and we cease iterations in the algorithm.

B. Estimation of MTTF, MGF and CDF

In order to find important statistical information from the lifetime data, we need to tie random variables Y and Y_d which can be established using Equations (50) and (55). In other words,

$$Y = ((c\theta^{(k^*)} + 1)^{1/\theta^{(k^*)}} - 1)\frac{Y_d - \min\{y_j^{(d)}\}}{\max\{y_j^{(d)}\} - \min\{y_j^{(d)}\}} + 1 \tag{56}$$

For notation simplicity, let $\alpha_{\theta^*}^{(d)} = (\max\{y_j^{(d)}\} - \min\{y_j^{(d)}\})/((c\theta^{(k^*)} + 1)^{1/\theta^{(k^*)}} - 1)$, then the expected value of Y_d can be calculated as

$$\mathbb{E}[Y_d] = \alpha_{\theta^*}^{(d)}\mathbb{E}[Y] - \alpha_{\theta^*}^{(d)} + \min\{y_j^{(d)}\} \tag{57}$$

Similarly, moment generating functions also satisfy the following relationship

$$M_{Y_d}(t) = e^{-(\alpha_{\theta^*}^{(d)} - \min\{y_j^{(d)}\})t} M_Y(\alpha_{\theta^*}^{(d)} t) \tag{58}$$

from which we can deduce all of the moments of the lifetime data (and hence the distribution of the lifetime data) provided the moments of Y, which is inverse-Box-Cox-transformed Argus distributed. Note that $M_Y(t)$ is given by (49) and $\mathbb{E}[Y^n]$ can be computed using either Equation (47) or (48) for $\theta > 0$ and $\theta = 0$, respectively. As for the density estimation, we state an important theorem from [56] below that establishes a simple relation between moment generating function and the cumulative distribution function.

Theorem 3.1 ([56])**.** *Let G be a continuous random variable with positive support and Λ be another independent random variable exponentially distributed with the expected value of 1. Let $T = \Lambda/G$, then*

$$P(T < t) = 1 - M_G(-t) \tag{59}$$

where $M_G(t)$ denotes the moment generating function of G.

Applying the result of this theorem (i.e., by setting $G = Y_d$) in our case, we will evaluate the CDF of Y_d as follows.

$$P(Y_d < t) = P(\frac{\Lambda}{T} < t) = P(\frac{\Lambda}{t} < T) = 1 - P(T < \frac{\Lambda}{t}) \tag{60}$$

where we can find $P(T < \Lambda/t)$ by conditioning on the random variable and then averaging as shown below

$$\begin{aligned} P(T < \frac{\Lambda}{t}) &= \int_0^\infty P(T < \lambda/t) f_\Lambda(\lambda) d\lambda & (61) \\ &= 1 - \int_0^\infty M_{Y_d}(-\lambda/t) f_\Lambda(\lambda) d\lambda & (62) \\ &= 1 - \int_0^\infty M_{Y_d}(-\lambda/t) e^{-\lambda} d\lambda & (63) \end{aligned}$$

Using equation (60) by plugging in equation (63), we can express the cumulative distribution function of Y_d as follows

$$\begin{aligned}
P(Y_d < t) =& \int_0^\infty M_{Y_d}(-\lambda/t)e^{-\lambda}d\lambda \\
=& \sum_{i=0}^{\infty} \frac{(-1)^i}{t^i i!}\mathbb{E}[Y_d^i]\int_0^\infty \lambda^i e^{-\lambda}d\lambda \\
=& \sum_{i=0}^{\infty} \frac{(-1)^i}{t^i}\mathbb{E}[Y_d^i] \qquad (64)\\
=& \sum_{i=0}^{\infty} \frac{(-1)^i}{t^i}\sum_{j=0}^{i}\binom{i}{j}(\alpha_{\theta^*}^{(d)})^{i-j} \\
& (\min\{y_j^{(d)}\} - \alpha_{\theta^*}^{(d)})^j\mathbb{E}[Y^{i-j}]
\end{aligned}$$

where the last equality follows from Equation (57) and the standard Binomial theorem. Finally, by taking derivative of $P(Y_d < t)$, we can calculate the probability density function estimate of Y_d as follows,

$$\begin{aligned}
f_{Yd}(t) =& \frac{d}{dt}P(Y_d < t) \\
=& \sum_{i=0}^{\infty} \frac{i(-1)^{i+1}}{t^{i+1}}\sum_{j=0}^{i}\binom{i}{j}(\alpha_{\theta^*}^{(d)})^{i-j} \\
& (\min\{y_j^{(d)}\} - \alpha_{\theta^*}^{(d)})^j\mathbb{E}[Y^{i-j}]
\end{aligned}$$

where the moments of Y i.e., $\mathbb{E}[Y^{i-j}]$ can be calculated by using either equations (47) or (48) based on the value of θ.

C. Non-Parametric Kernel Density Estimation

Before showing few numerical results, it is instructive to provide one of the classical non-parametric density evaluation techniques called KDE. In the classical approach to KDE, deriving useful statistics out of data does not involve any type of transformation i.e., it attempts to estimate the probability distribution directly by summing properly scaled kernel functions. KDE requires no prior knowledge of the underlying density function and automatically learns the shape of the density from the data. In our case, the estimate of Y_d would be given by

$$f_{Y_d}(y^{(d)}) = \frac{1}{Nh}\sum_{i=1}^{N}\mathcal{K}\left(\frac{y^{(d)} - y_j^d}{h}\right) \qquad (65)$$

where h is known as the smoothing bandwidth and $\mathcal{K}(.)$ is the kernel function. There are two key parameters to choose in a successful KDE. The choice of the kernel function $\mathcal{K}(.)$ and the smoothing bandwidth parameter. For a given kernel $\mathcal{K}(.)$, the optimal h that minimizes the asymptotic mean integrated square error is given by [57]

$$\begin{aligned}
h^* =& \arg\min_h \left(\frac{R(\mathcal{K})}{nh} + \frac{1}{4}\sigma_{\mathcal{K}}^4 R(f''_{Y_d}(y^{(d)}))h\right) \\
=& \left(\frac{R(\mathcal{K})}{N\sigma_{\mathcal{K}}^4 R(f''_{Y_d}(y^{(d)}))}\right)^{-0.2} \qquad (66)
\end{aligned}$$

where $f''_{Y_d}(.)$ denotes the second derivative and

$$R(L) = \int L^2(x)dx \text{ and } \sigma_{\mathcal{K}}^2 = \int x^2\mathcal{K}(x)dx. \qquad (67)$$

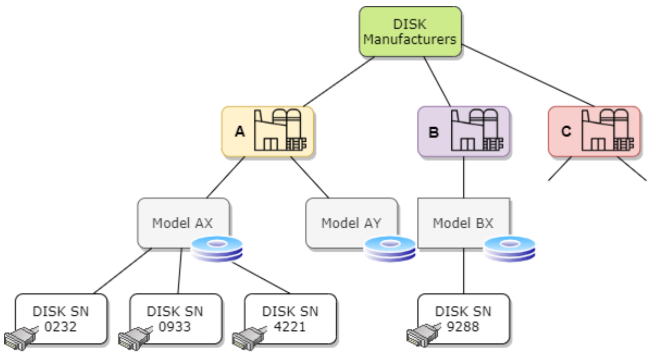


Fig. 3: Data classification over the BackBlaze data based on manufacturers and models. SN: Serial Number.

Unfortunately, the optimal h involves the unknown density function of Y_d which led the literature to look for data-based methods for selecting a suboptimal bandwidth parameter that will work well in practice [58], [15]. In addition it is shown in [59] that bandwidth estimation for KDE of heavy-tailed distributions is relatively a hard problem which requires additional generalized transformations and forms the basis of our motivation for this study. Given that the Box-Cox transformed data exhibits Argus-like distribution which is shown to have heavy tail, the proposed method in [59], which is not optimized for this particular distribution, will require additional complication in the density estimation process. Our scheme is comparably lower complexity which ensures good accuracy (and hence achieves a good trade-off point) as shall be demonstrated in the performance evaluation platform.

IV. Performance Evaluation Platform

A. Utilized BackBlaze Dataset

The data we use consists of around 80000 disk snapshots in the BackBlaze's data center collected over the last few years [50]. This data is stored as a database, including the date, serial numbers, model, capacity, working state, as well as SMART-based indicators. We have also added our dataset into Kaggle platform [60]. The statistical information may differ according to the manufacturer, model and serial numbers. In order to obtain a distribution, the data itself should be grouped together. An example grouping is shown in Fig. 3. In addition, capacity-based classifications can be made. Finally, within the use of SMART data, different model and manufacturer's disks can be used for our grouping and modeling operations.

B. Properties of Proposed Platform

The proposed platform interactively handles the collected hard drive data statistics of multiple disks which is pre-collected from multiple hard drive devices located in Backblaze the data center. Fig. 4 shows the general architecture of our platform solution so that proper analysis (such as MTTF calculation using the hard drive lifetime statistics) over the analytical framework can be performed. This structure consists of five basic modules: Data Source Layer, Data Collection Layer, (c) Data Processing and Storage Layer, (d) Visualization Layer, (e) Analysis Results.

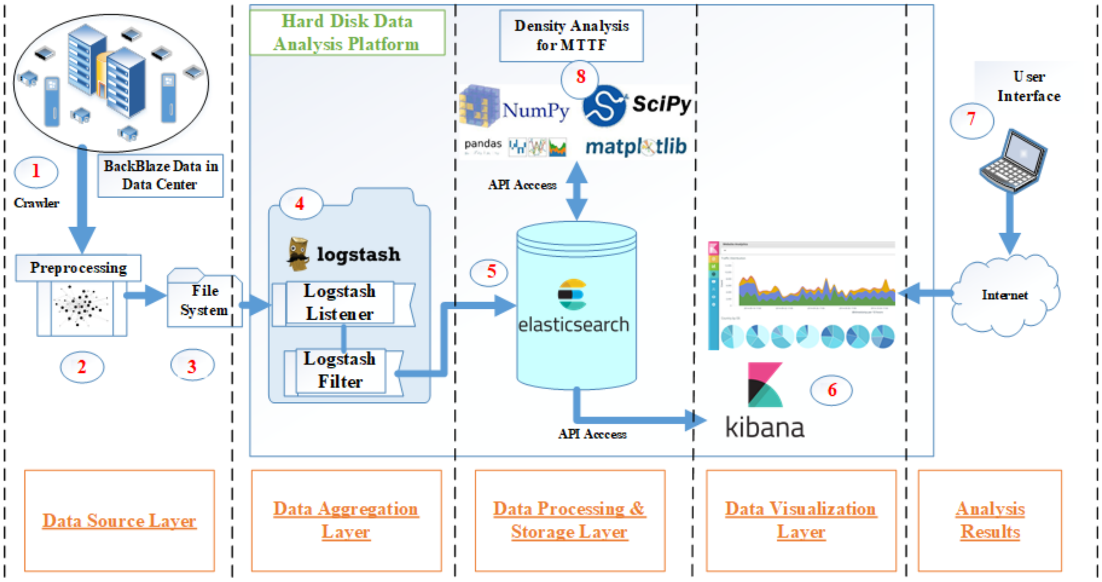


Fig. 4: Architectural structure that enables the stored data to be processed by the ElasticSearch framework

The proposed platform integrates data storage and search engine solutions with open source data analysis software (computational storage). After the data has been collected as in step-(1) from the data center that holds the hard drive data and their statistics as in Fig. 4, the data is processed sequentially inside the **Data Source Layer** marked as steps-(2) and (3). A pre-processing step is performed on the data as shown in step-(2) of Fig. 4. In the *Pre-Processing* phase, data summation such as aggregation of all data received in storage centers of various manufacturers, serial number, fault records, etc., filtering, is performed. After pre-processing, the data is stored in step-3 as a CSV file in a *File System*.

There is a Logstash component in the **Data Collection Layer** marked as step-(4) in Fig. 4 with two sub-components: *Logstash Listener* and *Logstash Transformer*. Logstash is a log parsing engine that performs logging, parsing and conversion operations [61]. *Logstash Listener* listens to the *file* generated in step-(3) and sends the records to *Logstash Transformer*. *Logstash Transformer* is used to convert the CSV data format of the *File System* to the ElasticSearch data format for later storage and analysis purposes. We use ElasticSearch, an open source, scalable full-text search and data analysis engine, in **Data Processing & Storage**, which is marked as step-(5) in Fig. 4. ElasticSearch enables query processing through structured and unstructured data storage. Inside data processing and storage layer, data enrichment, storage, access and analysis are executed. Hence, all computations regarding density analysis for fitting into the best PDFs, MTTF estimations (for failure prediction or classification purposes), Box-Cox transformations etc. are calculated in this layer. Elasticsearch indexes the data for further fast, reliable and distributed data processing purposes and provides simple RESTful Application Programming Interfaces (APIs). Using Python Elasticsearch Client API library [62], the density estimation and statistical analysis over the prior collected disk statistics data in ElasticSearch are done. The analysis results are later stored in ElasticSearch for further visualization purposes via Kibana. In the **Visualization Layer** marked as step-(6), we use Kibana, an analytical and visualization platform. In step-(7) marked in **Analysis Results Layer**, a user interacts with the platform via the platform's user interface. Pandas data analysis [63], which is marked as step-(8) and is in communication with ElasticSearch via API access in **Data Processing & Storage** Layer.

Note that system events (including SMART) have already been collected periodically for analysis and visualization in data centers today, and the common practice is to store the various logs in databases for easy queries. However, due to nature of the analyzed dataset that only includes meta-data information of storage failure statistics traditional approaches need to be extended for large scale analytics purposes. The proposed platform that includes data processing pipelines with data ingestion, analysis and process, storage and visualization layers can also be implemented with other existing open source tools such as Apache Hadoop, Spark, Pig, Hive, Kafka. For data ingestion, tools such as Apache Kafka, AWS Kinesis or RabbitMG, for data analysis and process tools such as Hadoop or AWS Map-Reduce, Apache Spark and AWS lambda, for data storage Hadoop HDFS, Hbase, Amazon S3 or DynamoDB and for data visualization Tableua, PowerBI can be some options. In this paper, we have chosen ElasticSearch stack which is designed as a scalable search engine to manage enormous volumes of diverse data, provides good performance for fast analytical query of the dataset that includes meta-data on storage system logs. Moreover, our analysis is based on building batch-data processing pipeline to capture, store, prepare, and analyze the failure storage data where ElasticSearch can be suitable option. ElasticSearch has also a large ecosystem of community.

V. Numerical Results

In this section, we numerically present our estimations of the hard disk lifetime distributions using the Backbaze's open

TABLE I: Analyzed disk Data Statistical Results

Total number of entries	54943404
Total number of failure records	2987
Failure percentage	% 0.0055
Average storage area (TB)	4.073
Average storage area of defective records (TB)	3.72
Number of different models	87
Defective number of different models	54
Average disk lifetime of defective models (days)	308.56
Total Duration of Records	2 years

dataset [50] and provide complexity and sensitivity analysis of the proposed platform.

A. Data Analytics Results

We use open source ElasticSearch data analytics tool to obtain the distributions of drive faults based on the models and manufacturers over time. ElasticSearch's Kibana visualization tool allows the visualization of disk properties in the system such as total disk capacity, usage details, failure figures and daily drive status information. In our analysis, we have ElasticSearch's query service to extract the lifetime of each disk model/series using Python libraries such as Pandas [63] and SciPY [64].

Table I provides an overview of the disk data statistics in the system. The analyzed data includes data between January 2016 and December 2017 (2 years) totalling up to around 55 million entries. Fig. 5(a) shows the comparison of the disk failure lifetime distributions between different and all brands using box-plot which includes Seagate (ST), Hitachi Global Storage Technologies (HGST), Hitachi, Western Digital Corporation (WDC) and Toshiba brands. Disk failure lifetime values are calculated using (1) by observing the elapsed time until a failure occurs. These values are extracted directly from the observed open source dataset. Fig. 5(a) shows that in the dataset, the disk failure lifetime of ST manufacturers is the highest with a median and mean values of 329 and 330.82 respectively, whereas the lifetime of Toshiba brand is the lowest with median and mean values of 74.50 and 111.31, respectively. Note that the median and mean lifetimes of all brands are 293 and 310.95, respectively. Furthermore, Fig. 5(b) shows the distribution of hard disk failure lifetime of all manufacturers as a histogram plot as well as the KDE applied where a high-tail characteristics of the distribution can clearly be observed.

Using real disk failure lifetime data, we have observed that the distribution of disk failure lifetime statistically follows a HT distribution. After our analysis involving distribution fitting, parameter estimations and subsequent quantile-quantile (Q-Q) plots, we obtain the best fit values using sample distributions including GHD, Argus and Exponential both before and after transforming the original hard disk data failure lifetime values via the Box-Cox function ($BC_\theta(y)$). For this transformation, *scaling* (s) and scrolling (translation) parameters *loc* (l) help to obtain the distribution using,

$$f_A(z;\chi,c) = f_A((y-l)/s;\chi,c)/s \qquad (68)$$

While making the choice of the fitted distribution, we have investigated 98 different continuous distributions given in Scipy statistical functions library [64]. In the best-fit evaluation process, we have used Sum Square Error (SSE) as the metric which is formulated as follows,

$$SSE = \sum_{i=1}^{N}(x_i - \hat{x}_i)^2, \qquad (69)$$

where N is the size of the bin, x_i is the actual observation and $\hat{x}_i$ is the estimate of the fitted distribution. In our numerical evaluations, we use CDF for actual and fitted observations. For comparisons of different PDFs, we have also utilized fit/test split where the dataset is split into fit and test datasets. In these calculations of SSE, the fitting distribution values $\hat{x}_i$ (using the relevant parameters of each distributions) are obtained using the fit dataset and the data values x_i are calculated from the test dataset. For fitting a probability distribution to data, we have utilized SciPy library in the hard disk data analysis platform in step-8 as given in Fig. 4. The fit function calculates the distribution of parameters that best matches to data (i.e. with the lowest level of error). In addition to Q-Q plots which are used to check how well the distribution fits into the data visually, we have also run a Kolmogorov-Smirnov (KS) test to measure goodness of fit using SciPy [64]. The overall evaluation methodology using the hard disk data analysis platform and how the best-fit parameters are obtained is summarized in Algorithm 1.

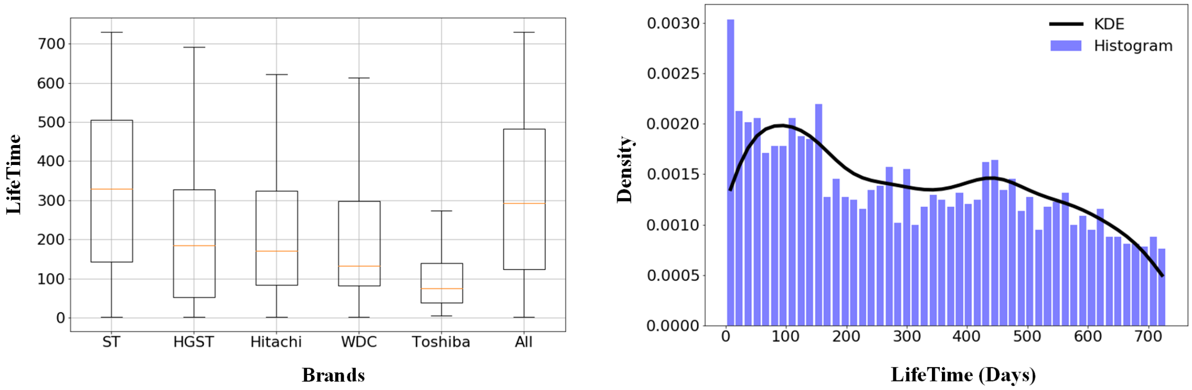
Algorithm 1: SSE evaluation methodology using best-fit parameters of the given PDFs.

```
Input: Pre-processed Backblaze open dataset, fit/test $split$ ratio, set
    of PDFs, $\mathcal{P} = \{1, 2, \ldots, P\}$
Output: $SSE(p), \ \forall p \in \mathcal{P}$
Initialization: Total number of iterations, $iter\_tot$
Split: Dataset into fit/test portions using $split$ ratio
foreach $n \in \{1, \ldots, iter\_tot\}$ do
    foreach $PDF{-}p \in \mathcal{P}$ do
        Fit: PDF$-p$ to dataset (fit portion) with SciPy's fit APIs
          [64]
        Obtain: The best fit parameters
        Build: CDF using the best fit parameters inside (68).
        Calculate: $SSE'(n,p)$ value between dataset (using test
          portion) and CDF using (69).
    end
end
Average: $SSE(p) = \sum_{n=1}^{iter\_tot} SSE'(n,p)$.
return $SSE(p)$
```

TABLE II: Estimated Argus values for MTTF (model), MTTF (Argus fit), MTTF (data) and MTTF (KDE)

Brands	MTTF (model)	MTTF (fit)	MTTF (data)	MTTF (KDE)
All	280.53	370.8	310.95	319.67

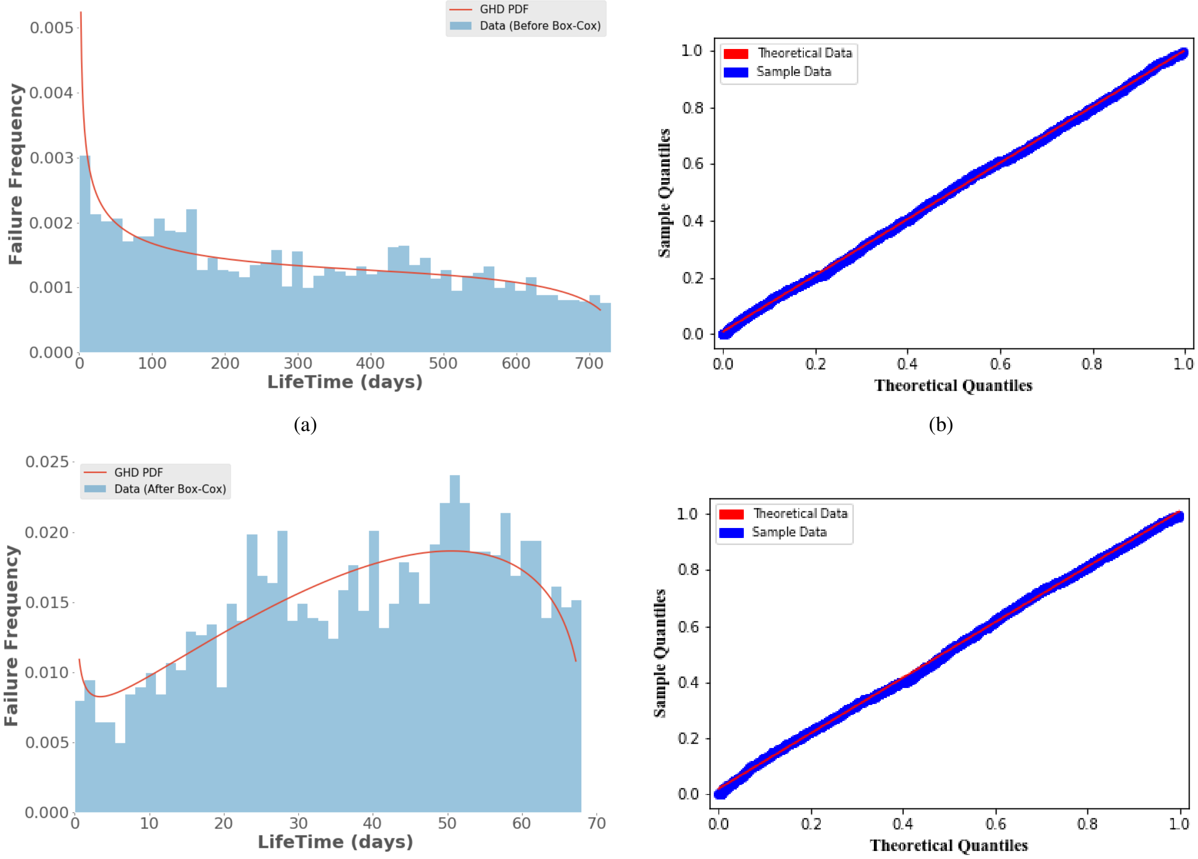
Fig. 6 presents the results of fitting the GHD (after scaling and shifting) over the hard-disk failure lifetime histogram data (both before and after Box-Cox transformation operation as shown in Fig. 6 (a) and Fig. 6 (c)), as well as the Q-Q diagrams which are obtained by taking the whole brands' disk lifetime into consideration as given in Fig. 6 (b) and Fig. 6 (d). From



(a)

(b)

Fig. 5: (a) Comparison of disk lifetime box-plots of different manufacturers (b) Distribution of hard disk failure lifetime with KDE plot.



(a)

(b)

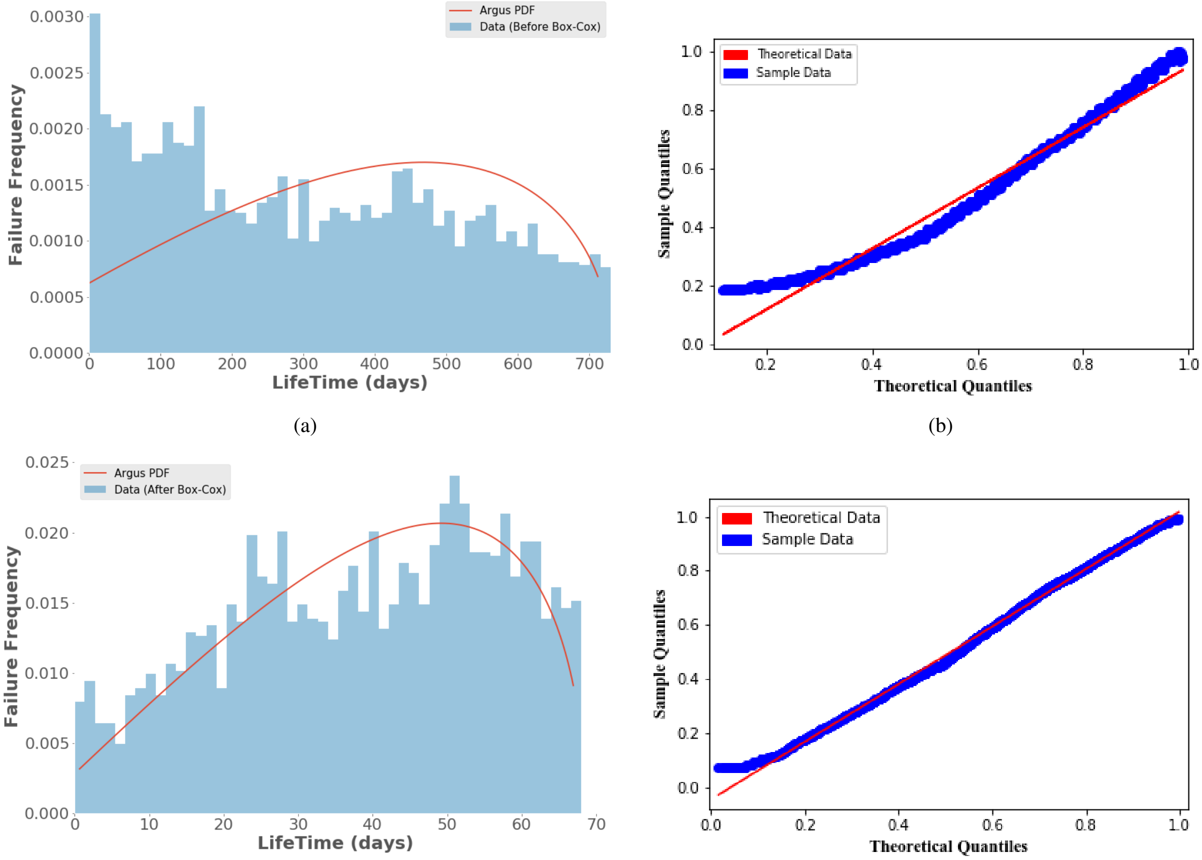
(c)

(d)

Fig. 6: Hard-disk failure lifetime with best fit distribution (a) GHD PDF v.s. the histogram of disk failure lifetime (before Box-Cox transformation) (b) Q-Q diagram (before Box-Cox transformation) (c) GHD PDF v.s. the histogram of disk failure lifetime (after Box-Cox transformation) (d) Q-Q diagram (after Box-Cox transformation)

these figures, we can easily observe that before and after Box-Cox transformations, the hard-disk failure lifetime data shows close fit to GHD.

Fig. 7(a) and Fig. 7(b) show the results of fitting the Argus distribution (after scaling and shifting) over the hard-disk failure lifetime data (before Box-Cox transformation), as well as the Q-Q diagram which is obtained by taking the whole brands' disk lifetime into consideration. We can observe that before Box-Cox transformation, the hard-disk failure lifetime data do not fit well to Argus distribution. Fig. 7(c) and Fig. 7(d), on the other hand demonstrate the Argus PDF v.s.



(a)

(b)

(c)

(d)

Fig. 7: Hard-disk failure lifetime with Argus distribution (a) Argus PDF v.s. the histogram of disk failure lifetime (before Box-Cox transformation) (b) Q-Q diagram (before Box-Cox transformation) (c) Argus PDF v.s. the histogram of disk failure lifetime (after Box-Cox transformation) (d) Q-Q diagram (after Box-Cox transformation)

TABLE III: Best fit parameters of GHD, Argus, Exponential and Weibull distributions and the corresponding MTTF values calculated by fitting each distribution into dataset.

All Brands	α	β	γ	z	λ	θ	χ	c_w	a	l	s	MTTF (fit)
GHD	0.72	1.32	1.65	-0.39	-	-	-	-	-	1	734	307.33
Argus	-	-	-	-	-	0.16	2e-4	-	-	-0.17	1.18	370.80
Exp.	-	-	-	-	32e-4	-	-	-	-	1	309.95	310.95
Weibull	-	-	-	-	-	-	-	1.30	333	0	1	307.76

the histogram of disk failure lifetime and the resulting Q-Q diagram after Box-Cox transformation. Fig. 7(d) shows that the data presents a better fit to Argus distribution after Box-Cox transformation despite some discrepancy at low quantiles. The Box-Cox transformation uses the parameter $\theta = 0.5545$ that is selected to maximize the associated log-likelihood function. Note that with these values, it can be verified that we satisfy the prementioned condition $|\theta(sc + l)| \leq 1$ in the text. Table II summarizes the fit parameters of the Argus distribution as well as the calculated values of the MTTF (using values obtained from equation (57), fitted to Argus distribution, dataset and KDE methods) for all disk manufacturers combined. As the amount of data increases, we can obtain better MTTF estimations.

Fig. 8 shows the hard-disk failure lifetime with Exponential distribution fitting before and after Box-Cox transformations. We can observe from these figures that the hard-disk failure lifetime does not fit to a simple distribution such as exponential i.e., there is large discrepancy over large quantile ranges. Finally, Fig. 9 shows the dataset fitting results into Weibull distribution before and after Box-Cox transformation. Similar to exponential fit results, we can observe that the dataset do not fit well into the Weibull distribution (especially at high quantiles) with the best fitting parameters as given in Table III.

Table III summarizes the fit parameters of GHD, Exponential, Argus, and Weibull distributions as well as the corresponding calculated MTTF values for each distribution. We have used *Scipy* library to do all of our regression work. The mean values, which corresponds to MTTF (fit) values in

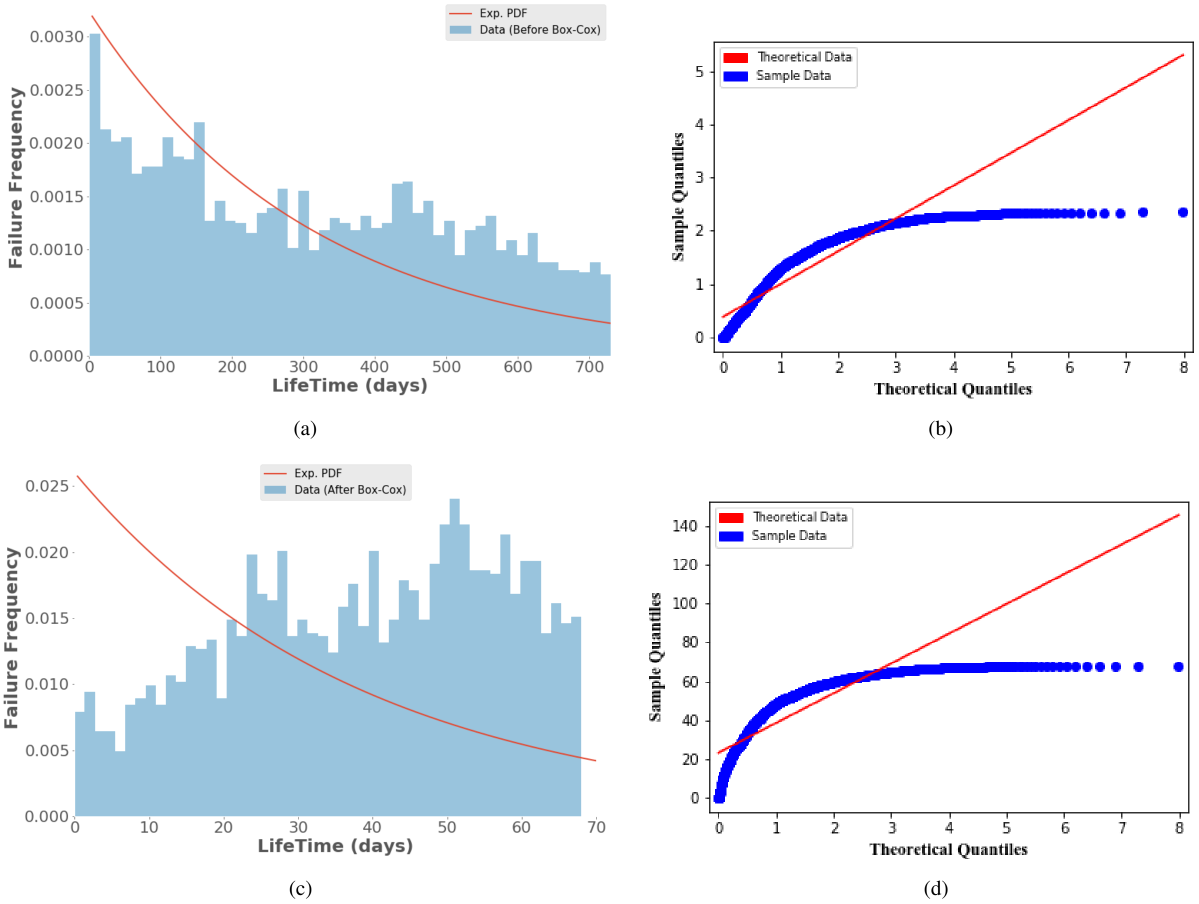


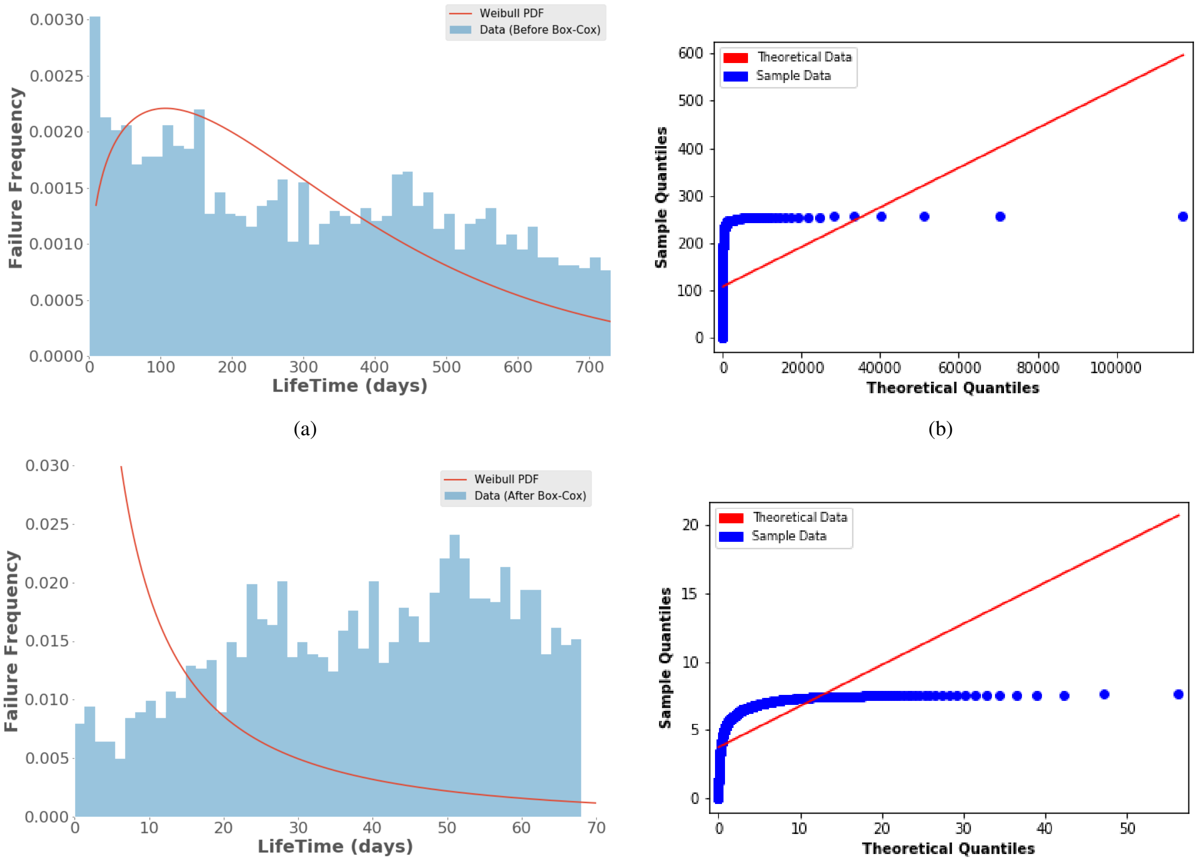
Fig. 8: Hard-disk failure lifetime with Exponential distribution (a) Exponential PDF v.s. the histogram of disk failure lifetime (before Box-Cox transformation) (b) Q-Q diagram (before Box-Cox transformation) (b) Exponential PDF v.s. the histogram of disk failure lifetime (after Box-Cox transformation) (c) Q-Q diagram (after Box-Cox transformation)

Table III, are 371, 307, 311 and 307 days for Argus, GHD, Exponential and Weibull distributions respectively. Notice that with smaller estimate compared to the data mean, GHD might be overfitting. Note also that the best fitting shape parameter in a Weibull distribution is calculated to be $c_w = 1.30$ and since $c_w > 1.0$ the HDD failure rate is observed to be increasing [31].

Fig. 10(a) and Fig. 10(b) show the CDF plots of all models including GHD, Argus, Exponential, Weibull distributions as well as KDE by fitting to dataset of Table I before and after Box-Cox transformation. First of all, before Box-Cox transformation as shown in Fig. 10(a) among all the distributions, we can observe that GHD distribution and KDE yield better fit to dataset than other distributions. This can also be validated with the SSE values where in terms of SSE values, GHD and KDE distributions yield the lowest SSE values with 0.06 and 0.02 respectively. Second, after Box Cox transformation as shown in Fig. 10(b) among all the distributions, GHD and KDE distributions show better fit to the Box-Cox transformed dataset. In this case, the SSE values for GHD and KDE distributions are around 0.05 and 0.01 respectively. Finally, Fig. 10(a) depicts that Argus distribution yields higher lifetime estimations compared to other distributions. For example, for a value of 0.5 on y-axis (corresponding to median values), Argus distribution yields 376 days, GHD yields 288 days, Exponential distribution yields 216 days and Weibull distribution yields 293 days. However, note that GHD achieves this middle ground by using four different optimized parameters (with the potential risk of overfitting) whereas Argus uses only two parameters.

Finally as a side note, Fig. 11 shows the evolution of α and MTTF values over time of our iterative algorithm given for Argus distribution in Fig. 2, for all brands, ST and HGST in particular. We also observe that convergences are achieved with only few iterations validating the low complexity nature of the proposed iterative scheme. More specifically, Fig. 11(a) shows that for all brands, ST and HGST, MTTF convergence is achieved in 4, 3 and 5 iterations respectively using (57). Fig. 11 (b) shows that for all brands, ST and HGST, θ convergence is achieved in 8, 5 and 53 iterations respectively.

Table IV shows the KS test results to evaluate the goodness of fit for each considered distributions [65]. KS test is used to measure the goodness of fit between the collected dataset and the considered models to find their similarity. It then outputs a distance value that shows the maximum difference between CDFs. Using KS test, we obtained both KS distance statistic and p-value. Table IV shows that Argus-data KS distance is



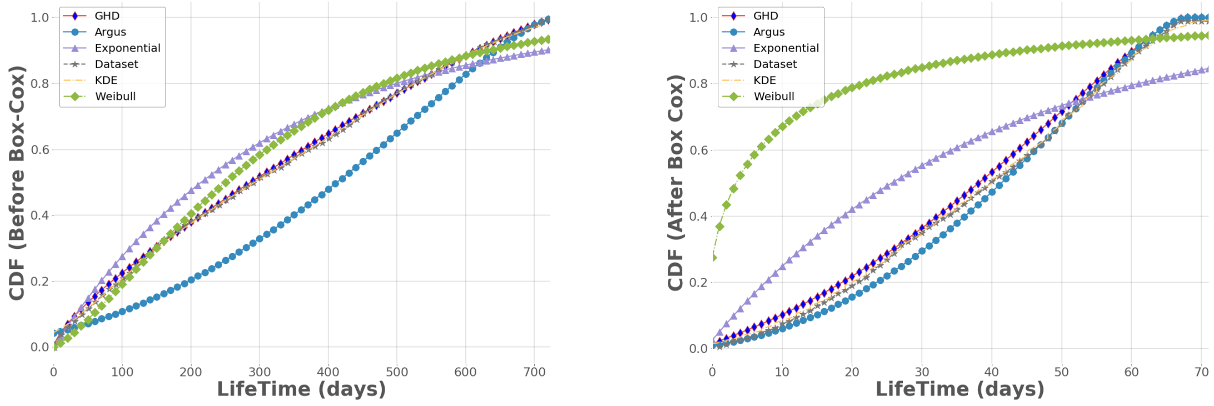
(a)

(b)

(c)

(d)

Fig. 9: Hard-disk failure lifetime with Weibull distribution (a) Weibull PDF v.s. the histogram of disk failure lifetime (before Box-Cox transformation) (b) Q-Q diagram (before Box-Cox transformation) (b) Weibull PDF v.s. the histogram of disk failure lifetime (after Box-Cox transformation) (c) Q-Q diagram (after Box-Cox transformation)

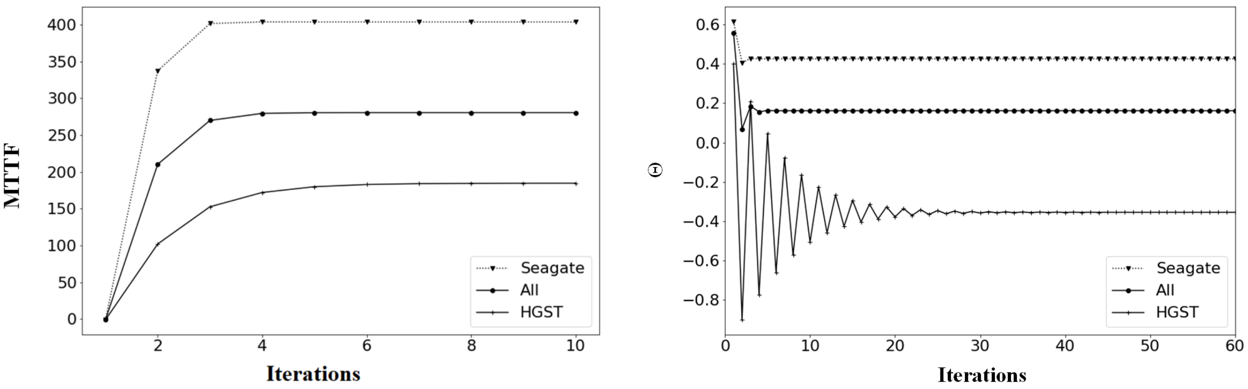


(a)

(b)

Fig. 10: (a) CDF plots of all distributions and MTTF extracted from dataset before Box-Cox. (b) CDF plots of all distributions and MTTF extracted from dataset after Box-Cox.

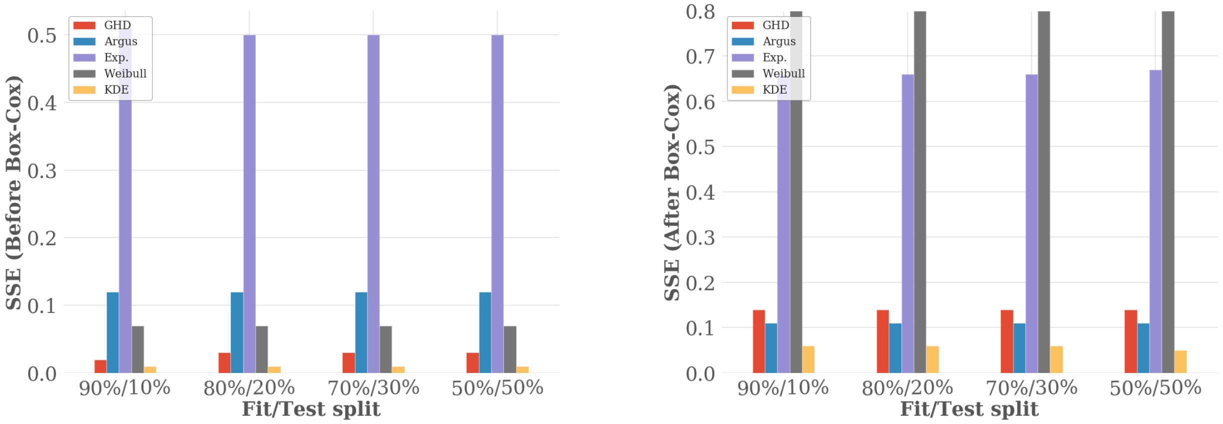
again observed to be the smallest distance (with a value of 0.07) after Box-Cox transformation for all fit/test split ratios. The p-values are also observed to be around zero suggesting that the difference between the two distributions is significant with a near certainty.



(a)

(b)

Fig. 11: (a) MTTF and (b) θ values as a function of iterations illustrating the MTTF convergence curve for Argus distribution.



(a)

(b)

Fig. 12: Plots of fit/test split ratio versus SSE for various pdfs (a) Before Box-Cox. (b) After Box-Cox.

TABLE IV: Comparisons of PDFs and dataset in terms of KS distance and p values

Box-Cox	**Fit/Test split ratio**	**GHD-data (KS dist., p-value)**	**Argus-data (KS dist., p-value)**	**Exp.-data (KS dist., p-value)**	**Weibull-data (KS dist., p-value)**
Before	90%/10%	(0.07,0)	(0.29,0)	(0.22,0)	(0.10,0)
After		(0.08,0)	(0.07,0)	(0.25,0)	(0.86,0)
Before	80%/20%	(0.07,0)	(0.31,0)	(0.22,0)	(0.10,0)
After		(0.08,0)	(0.07,0)	(0.25,0)	(0.86,0)
Before	70%/30%	(0.08,0)	(0.20,0)	(0.22,0)	(0.10,0)
After		(0.08,0)	(0.07,0)	(0.25,0)	(0.86,0)
Before	50%/50%	(0.07,0)	(0.25,0)	(0.22,0)	(0.10,0)
After		(0.08,0)	(0.07,0)	(0.25,0)	(0.86,0)

B. Complexity and Sensitivity Analysis

Although providing a detailed complexity analysis is beyond the scope of this paper, we would like to indicate the connection between parameter size of the underlying distribution and the complexity of fitting. As the model complexity (the number of parameters and the distribution) increases, the goodness of fitting increases [66]. However, the increase is not linear but rather concave giving us diminishing returns. We also see the similar trends in our case.

Let us also present the shuffled data on the SSE of the distributions over a bigger data set this time. This data set contains all the failures (a total of $162,649$ failures of unique serial numbers) during the time period between april 10th, 2013 to march 31st, 2019. We have run our experiments $iter\_tot = 100$ times and SSE averages are presented in Fig. 12 before and after Box-cox transformation. We have included SSE values of different distributions when the dataset is split into fit and test samples indicated by the fit/test split ratios. Over all fit/test data split cases, before Box Cox transformation, we can observe that KDE performs the best followed by GHD, Exponential and Argus PDFs. Note that KDE uses a combination of multiple Gaussian distributions (instead of single one) to achieve this performance and it is included in our table for reference purposes. After Box-Cox transformation, again KDE performs the best but this time it is followed by Argus, GHD and Exponential PDFs respectively. Note that Argus distribution yields smaller SSE values compared to GHD after Box-Cox transformation. This is due to the fact that Argus distribution does not overfit the data and yields better performance on the unseen test dataset (achiving the minimal generalization error). This validates our assumption that Argus distribution can give better bias/variance trade-off compared to other complex distributions such as GHD which are likely to overfit the model to the fit dataset with more complex operation. Finally, as the fit/test split decreases to $50\%/50\%$, no significant change can be observed on SSE gap values.

VI. Conclusions and Future Work

In this paper, we have studied the probabilistic modeling of real-life disk failure lifetime as well as analyzed the storage statistics. Contrary to classical approaches such as KDE, we derived novel and alternative methods to find known statistical metrics about the lifetime and hence the reliability such as MTTF (in a closed form) for non-repairable systems as well as the density estimation of the transformed input data through MGFs. Our thorough analysis demonstrates that the hard disk failure lifetime distribution of the observed dataset closely follows GHD and that Box-Cox transformed data closely follows an Argus-like distribution up to a proper scaling and shifting which achieves a good bias/variance trade-off. To this end, we have also considered/compared other distributions such as exponential (single-parameter estimation), Weibull (two-parameter estimation) as well as GHD (four-parameter estimation) functions to fit to data with and without transformations. Our proposed scheme illustrates a practical data utilization strategy and analytically derive closed form expressions to estimate the reliability which in our opinion would be quite valuable for reliability analysts as well as engineers to quickly get a handle on their system. Note that our scheme do not use disk-specific health monitoring data (such as SMART). We basically focus on more basic indicators such as lifetime. Hence, we believe that our study would directly be applicable to SSDs drive data without much of a major change. For instance, the bit error predictive modelling of [67] can be thought in accordance with an appropriate error correction coding mechanism to estimate the failure times of such devices such as a threshold error rate beyond which the data is not longer recoverable. Thus, this time period can be considered as an input to our prediction model to accurately estimate the corresponding lifetime. In our future work, we ar e planning to include the SMART data that shall complement our failure error prediction model proposed in this study to dynamically supplement the decision mechanism real-time. The results of this study will be more like fitting data to resolve the relationship between SMART data and the lifetime through the known deep learning techniques.

Acknowledgment

This work was partially funded by The Scientific and Technological Research Council of Turkey (TUBITAK) under the grant numbers 115C111 and 119E235, Spanish MINECO under the grant number TEC2017-88373-R (5G-REFINE) and by Generalitat de Catalunya under the grant number 2017SGR1195.

References

[1] B. Schroeder and G. A. Gibson, “Disk failures in the real world: What does an mttf of 1 000 000 hours mean to you?” in *in Proc. 5th USENIX Conf. FAST*, 2007, pp. 1–16.

[2] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[3] B. A. Turlach, “Bandwidth selection in kernel density estimation: A review,” in *CORE and Institut de Statistique*. Citeseer, 1993.

[4] X. Zhu and V. Melnykov, “Manly transformation in finite mixture modeling,” *Computational Statistics & Data Analysis*, vol. 121, pp. 190–208, 2018.

[5] M. S. Rothberg, “Disk drive for receiving setup data in a self monitoring analysis and reporting technology (smart) command,” May 17 2005, uS Patent 6,895,500.

[6] E. Pinheiro, W.-D. Weber, and L. A. Barroso, “Failure trends in a large disk drive population.” in *FAST*, vol. 7, no. 1, 2007, pp. 17–23.

[7] I. C. Chaves, M. R. P. d. 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 *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, Oct 2016, pp. 427–432.

[8] B. Vineetha and A. Borah, “Enhanced rules framework for predicting disk drives failures,” *International Journal of Computer Science and Mobile Computing*, vol. 5, no. 7, pp. 177–185, 2016.

[9] P. Anantharaman, M. Qiao, and D. Jadav, “Large scale predictive analytics for hard disk remaining useful life estimation,” in *2018 IEEE International Congress on Big Data (BigData Congress)*, July 2018, pp. 251–254.

[10] S. Basak, S. Sengupta, and 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)*, June 2019, pp. 208–216.

[11] S. Durasevic and B. Dordevic, “Anomaly detection using SMART indicators for hard disk drive failure prediction,” in *4th International Conference on Electrical, Electronics and Computing Engineering (IcETRAN 2017)*, 2017, pp. 1–4.

[12] Y. X. et al, “Improving service availability of cloud systems by predicting disk error,” in *USENIX Annual Technical Conference (USENIX ATC 18)*, 2018, pp. 481–494.

[13] J. X. et al, “Disk failure prediction in data centers via online learning,” in *Proceedings of the 47th International Conference on Parallel Processing*, 2018, pp. 1–10.

[14] Y. Xie, D. Feng, F. Wang, X. Zhang, J. Han, and X. Tang, “Ome: An optimized modeling engine for disk failure prediction in heterogeneous datacenter,” in *2018 IEEE 36th International Conference on Computer Design (ICCD)*, Oct 2018, pp. 561–564.

[15] X. Xu, Z. Yan, and S. Xu, “Estimating wind speed probability distribution by diffusion-based kernel density method,” *Electric Power Systems Research*, vol. 121, pp. 28–37, 2015.

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

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

[18] S. S. Arslan, “Redundancy and aging of efficient multidimensional MDS parity-protected distributed storage systems,” in *IEEE Trans. Device Mater. Rel., vol. 14, no. 1*, 2014, pp. 275–285.

[19] S. S. Arslan, “A reliability model for dependent and distributed mds disk array units,” *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 133–148, March 2019.

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

[21] N. Aussel, S. Jaulin, G. Gandon, Y. Petetin, E. Fazli, and S. Chabridon, "Predictive models of hard drive failures based on operational data," in *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. IEEE, 2017, pp. 619–625.

[22] M. M. Botezatu, I. Giurgiu, J. Bogojeska, and D. Wiesmann, "Predicting disk replacement towards reliable data centers," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 39–48.

[23] C. C. A. Rincón, J.-F. Pâris, R. Vilalta, A. M. Cheng, and D. D. Long, "Disk failure prediction in heterogeneous environments," in *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2017 International Symposium on*. IEEE, 2017, pp. 1–7.

[24] A. R. Mashhadi, W. Cade, and S. Behdad, "Moving towards real-time data-driven quality monitoring: A case study of hard disk drives," *Procedia Manufacturing*, vol. 26, pp. 1107–1115, 2018.

[25] F. L. F. Pereira, F. D. dos Santos Lima, L. G. de Moura Leite, J. P. P. Gomes, and J. de Castro Machado, "Transfer learning for bayesian networks with application on hard disk drives failure prediction," in *Intelligent Systems (BRACIS), 2017 Brazilian Conference on*. IEEE, 2017, pp. 228–233.

[26] F. D. dos Santos Lima, G. M. R. Amaral, L. G. de Moura Leite, J. P. P. Gomes, and J. de Castro Machado, "Predicting failures in hard drives with lstm networks," in *2017 Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2017, pp. 222–227.

[27] E. T. Mittman, C. Lewis-Beck, and W. Q. Meeker, "A hierarchical model for heterogenous reliability field data," *Technometrics*, 2018.

[28] E. Zeydan and S. S. Arslan, "Cloud2HDD: Large-Scale HDD Data Analysis on Cloud for Cloud Datacenters," in *2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2020, pp. 243–249.

[29] L. N. Bairavasundaram, G. R. Goodson, S. Pasupathy, and J. Schindler, "An analysis of latent sector errors in disk drives," in *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 2007, p. 289–300.

[30] K. M. G. et al, "Mean Time to Meaningless: MTTDL, Markov Models, and Storage System Reliability." in *HotStorage*, 2010, pp. 1–5.

[31] J. G. Elerath and M. Pecht, "Enhanced reliability modeling of raid storage systems," in *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*, June 2007, pp. 175–184.

[32] J. Pâris, S. J. T. Schwarz, S. J. A. Amer, and D. D. E. Long, "Protecting raid arrays against unexpectedly high disk failure rates," in *2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing*, Nov 2014, pp. 68–75.

[33] M. Kishani and H. Asadi, "Modeling impact of human errors on the data unavailability and data loss of storage systems," *IEEE Transactions on Reliability*, vol. 67, no. 3, pp. 1111–1127, 2018.

[34] S. Kadekodi, K. Rashmi, and G. R. Ganger, "Cluster storage systems gotta have heart: improving storage efficiency by exploiting disk-reliability heterogeneity," in *17th {USENIX} Conference on File and Storage Technologies ({FAST} 19)*, 2019, pp. 345–358.

[35] V. Venkatesan and I. Iliadis, "A general reliability model for data storage systems," in *2012 Ninth International Conference on Quantitative Evaluation of Systems*. IEEE, 2012, pp. 209–219.

[36] Z. Qiao, J. Hochstetler, S. Liang, S. Fu, H.-b. Chen, and B. Settlemyer, "Developing cost-effective data rescue schemes to tackle disk failures in data centers," in *International Conference on Big Data*. Springer, 2018, pp. 194–208.

[37] J. G. Elerath and S. Shah, "Server class disk drives: how reliable are they?" in *Annual Symposium Reliability and Maintainability, 2004 - RAMS*, Jan 2004, pp. 151–156.

[38] S. Shah and J. G. Elerath, "Reliability analysis of disk drive failure mechanisms," in *Annual Reliability and Maintainability Symposium, 2005. Proceedings.*, Jan 2005, pp. 226–231.

[39] J. G. Elerath, "Specifying reliability in the disk drive industry: No more mtbf's," in *Annual Reliability and Maintainability Symposium. 2000 Proceedings. International Symposium on Product Quality and Integrity (Cat. No.00CH37055)*, Jan 2000, pp. 194–199.

[40] J. Elerath and M. Pecht, "A highly accurate method for assessing reliability of redundant arrays of inexpensive disks (raid)," *IEEE Transactions on Computers*, vol. 58, no. 3, pp. 289–299, 2008.

[41] J. Elerath, "Hard-disk drives: The good, the bad, and the ugly," *Communications of the ACM*, vol. 52, no. 6, pp. 38–45, 2009.

[42] J. G. Elerath and S. Shah, "Disk drive reliability case study: dependence upon head fly-height and quantity of heads," in *Annual Reliability and Maintainability Symposium.*, Jan 2003, pp. 608–612.

[43] S. Shah and J. G. Elerath, "Disk drive vintage and its effect on reliability," in *Annual Symposium Reliability and Maintainability, RAMS*, Jan 2004, pp. 163–167.

[44] S. Shah and J. G. Elerath, "Method and system for reliability analysis of disk drive failures," November 2006, uS Patent 7,136,768.

[45] J. G. Elerath and J. Schindler, "Beyond MTTDL: A closed-form RAID 6 reliability equation," *ACM Transactions on Storage (TOS)*, vol. 10, no. 2, pp. 1–21, 2014.

[46] J. G. Elerath, "A simple equation for estimating reliability of an n+1 redundant array of independent disks (raid)," in *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, June 2009, pp. 484–493.

[47] T. Zhang and B. Yang, "Box–cox transformation in big data," *Technometrics*, vol. 59, no. 2, pp. 189–201, 2017. [Online]. Available: https://doi.org/10.1080/00401706.2016.1156025

[48] Y. Gao, T. Zhang, and B. Yang, "Finding the best box-cox transformation in big data with meta-model learning: A case study on qct developer cloud," in *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, June 2017, pp. 31–34.

[49] J. Rodríguez-Avi, A. Conde-Sánchez, A. Sáez-Castillo, and M. Olmo Jiménez, "Estimation of parameters in gaussian hypergeometric distributions," *Communications in Statistics-Theory and Methods*, vol. 32, no. 6, pp. 1101–1118, 2003.

[50] Backblaze, "Hard drive data and stats," https://www.backblaze.com/b2/hard-drive-test-data.html, 2018, [Online; accessed 31-October-2019].

[51] S. Foss, D. Korshunov, S. Zachary, *et al.*, *An introduction to heavy-tailed and subexponential distributions*. Springer, 2011, vol. 6.

[52] "Gauss Hypergeometric Distribution," https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gausshyper.html#scipy.stats.gausshyper, 2017, [Online; accessed 18-March-2019].

[53] W. Gong, Y. Liu, V. Misra, and D. Towsley, "On the tails of web file size distributions," in *Proceedings of the annual allerton conference on communication control and computing*, vol. 39, no. 1. The University; 1998, 2001, pp. 192–201.

[54] L. Oliveira, K. Obraczka, and A. Rodríguez, "Characterizing user activity in wifi networks: University campus and urban area case studies," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems: Malta, Malta*, 2016.

[55] P. Li, "Box-cox transformations: an overview," *presentation, http://www. stat. uconn. edu/˜ studentjournal/index_files/pengfi_s05. pdf*, 2005.

[56] A. Rossberg, "Laplace transforms of probability distributions and their inversions are easy on logarithmic scales," *Journal of Applied Probability*, vol. 45, no. 2, pp. 531–541, 2008.

[57] M. C. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," *Journal of the american statistical association*, vol. 91, no. 433, pp. 401–407, 1996.

[58] N. Agarwal and N. Aluru, "A data-driven stochastic collocation approach for uncertainty quantification in mems," *International Journal for Numerical Methods in Engineering*, vol. 83, no. 5, pp. 575–597, 2010.

[59] T. Buch-Larsen, J. P. Nielsen, M. Guillén, and C. Bolancé, "Kernel density estimation for heavy-tailed distributions using the champernowne transformation," *Statistics*, vol. 39, no. 6, pp. 503–516, 2005.

[60] Kaggle Dataset Platform, "Dataset: Hard Drive Failure Statistics," https://www.kaggle.com/ezeydan/hard-drive-failure-data, 2019, [Online; accessed 15-April-2019].

[61] "Logstash: Collect, Parse, Transform Logs — Elastic," https://www.elastic.co/products/logstash, 2017, [Online; accessed 11-Sept.-2017].

[62] H. Král, "Python Elasticsearch Client," https://elasticsearch-py.readthedocs.io/en/master/index.html, 2013, [Online; accessed 12-Feb.-2020].

[63] "Python Data Analysis Library," https://pandas.pydata.org/, 2018, [Online; accessed 22-Jan.-2018].

[64] "Statistical functions (scipy.stats)," https://docs.scipy.org/doc/scipy/reference/stats.html, 2019, [Online; accessed 15-April-2019].

[65] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[66] I. V. Cadez and P. Smyth, "Model complexity, goodness of fit and diminishing returns," in *Advances in Neural Information Processing Systems*, 2001, pp. 388–394.

[67] D. Wei, L. Qiao, X. Chen, H. Feng, and X. Peng, "Prediction models of bit errors for nand flash memory using 200 days of measured data," *Review of Scientific Instruments*, vol. 90, no. 6, p. 064702, 2019.