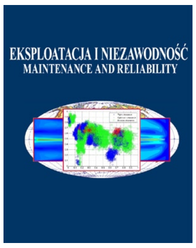

Eksploatacja i Niezawodnosc – Maintenance and Reliability

Volume 24 (2022), Issue 2

journal homepage: http://www.ein.org.pl

Article citation info:
Baranowski J. Predicting IoT failures with Bayesian workflow. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2022; 24 (2): 248–259, http://doi.org/10.17531/ein.2022.2.6.

Predicting IoT failures with Bayesian workflow

Indexed by:
Web of Science Group

Jerzy Baranowski[a]

[a]AGH University of Science & Technology; Al. A. Mickiewicza 30, 30-059 Kraków, Poland

Highlights

- We consider predicting failure time of IoT devices in networks.
- We propose a data fusion of happenstance data.
- Proposed Bayesian hierarchical model captures between group variance.
- Post-stratification helps in generalization.

Abstract

IoT networks are so voluminous that they cannot be treated as individual devices, but as populations. Main aim of the paper is to create a comprehensive method for predicting failures taking device variance into consideration. We propose using data fusion of happenstance observations (resets and failures) to better estimate device parameters. We propose using methods of population analysis in Bayesian statistics to estimate failure times investigating only a part of the population. For this purpose, we use multilevel hierarchical Bayesian model and provide it with post stratification. We propose model assumptions, construct the model and evaluate it, and perform computations using Hamiltonian Monte Carlo. This method is known as the Bayesian workflow. We have analyzed three different models showing that, in case of small device variance, it can be ignored, or at least compensated, while significant differences require hierarchical modeling. We also show that hierarchical model shows significant robustness to a small amount of data. We have shown attractiveness of Bayesian approach to modeling failures of IoT devices. Ability to diagnose and tune models, and assure their computational fidelity is a great advantage of Bayesian workflow.

This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/)

Keywords

IoT failures; Bayesian workflow; post-stratification; Hamiltonian Monte Carlo.

1. Introduction

We can say, that Internet-of-Things (IoT) devices have become a ubiquitous aspect of current industry and life. The remote monitoring systems using Internet of Things include vehicle or assets monitoring, people/pets monitoring, fleet management, water and oil leakage, energy grid and power plant monitoring etc. Main applications of IoT devices include on-line monitoring and predictive maintenance of industrial equipment. Their use provides both process monitoring for constant quality assurance, and condition monitoring in order to prevent unplanned downtimes. The machines and devices diagnostics, their maintenance and methods of preventing failures with the use of IoT relate to various fields such as Industry 4.0, management of transport devices or medical devices.

Currently rising in relevance and especially interesting field are the IoT networks comprising multiples of same device. Example would be an industrial plant sensor network, which collects data regarding the operation of the entire installation at various points. Failures of sensor devices are not uncommon, especially if the devices are cheap and/or mass produced. This, of course, causes degradation of coverage, but also introduces a multitude of problems. Replacement of networked devices during plant operation might be difficult – for example, because of physical accessibility. Usual reasons for switching to wireless solutions are difficulties with providing necessary cable connections. IoT device failures introduce difficulties in energy management, while they are usually low power solutions, their number complicates matters. This is even more troublesome for battery operated devices.

As IoT networks are not failure free, we need to consider accidental failures, such as broken communication links or failures of cyber instances and cyber-attacks. Even sensors made in the same factory can have significant variance. Thus ensuring the robustness of IoT devices and predicting the failure of these devices is a very important issue. In this subject, we can divide research studies into the following categories:

- IoT devices anomaly detection,
- attack detection in IoT,
- self-healing IoT sensors,
- cascade-of-failures across domains.

(*) Corresponding author.

E-mail addresses: J. Baranowski (ORCID: 0000-0003-3313-581X): jb@agh.edu.pl

Because of advances in sensor monitoring technology, low-cost solutions, and high impact in various application domains, IoT systems are becoming increasingly popular, and anomaly detection has attracted considerable attention from the research community Fahim and Sillitti [9] present a literature review of anomaly detection in IoT systems. They identify several research gaps related to data collection, analysis of unbalanced large data sets, limitations of statistical methods in processing massive sensory data. They note that a few research articles deal with predicting anomalous behavior in real-world scenarios. In anomaly recognition, the focus of most papers is on automatic detection of anomalies in data. Key considerations are point, contextual, and collective anomalies. Rarely, they focus on relationships of data anomalies with sensor failures. Kaya et al. [15] present the prediction of sensor failure because of aging or environmental factors in the food industry. They base proper food quality assessment on the correct operation of the sensors used. The authors examine the loss of data, in order to tolerate for the failed sensor and keep the overall prediction accuracy acceptable. They propose a Single Plurality Voting System (SPVS) classification approach, applying K-Nearest Neighbor (kNN), Decision Tree, and Linear Discriminant Analysis (LDA) as the base classifiers.

Vangipuram et al. [23] present detection of missing values in IoT environment based on open sourced data. The authors presented a technique for imputation of missing data values, a classifier that is based on feature transformation to perform classification, and an imputation measure to compute the similarity between any two instances, also useful as a similarity measure. They tested the performance of the proposed classifier using imputed datasets got by applying Kmeans, F-Kmeans and the proposed imputation methods. Moghaddam and Muccini [20] present an overview of methods, techniques and architectures for Fault-tolerance in IoT. Failure can occur at all levels of the Internet of Things (IoT) application architecture, e.g., oversight of sensor nodes, failure of network links, and failure of components that process and store data. For this reason, fault-tolerance (FT) has become a key issue for IoT systems.

Anomaly detection and prediction is the first step to secure IoT systems. The goal of IoT security is to protect assets, ensure the privacy, confidentiality, availability, and integrity of communications in the IoT ecosystem. Therefore, IoT security has recently gained the attention of researchers in attack detection. Hasan et al. [14] analyzed performances of several machine learning models and compared them in attack and anomaly prediction on the IoT systems accurately. The authors consider several machine learning (ML) algorithms, such as Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Artificial Neural Network (ANN). Diro and Chilamkurti [8] present the application of deep learning to enable attack detection in the social web. The authors compare a deep and a shallow neural network using open source dataset and show that the deep model is more effective in detecting attacks than its shallow counterparts. Manimurugan et al. [18] also consider deep machine learning for anomaly and intrusion detection in IoT. They propose a model of the Deep Belief Network (DBN) algorithm.

Self-healing IoT sensors are the next research area. Lin et al. [16] propose a solution called SensorTalk for automatic detection of potential sensor failures and calibrate the aging sensors semi-automatically. They have proposed both analytical and simulation models for the detection delay selection so that when a potential failure occurs; it detects it early enough without causing too many false alarms. Cascading failures are another important topic. In the Internet of Things (IoT), various devices work together to collect data, communicate information to each other, and process it intelligently. Because of the interactions and dependencies between IoT devices, a malfunction of one of them can trigger a cascade of failures. Xing [27] systematically reviews cascading failure modeling and reliability analysis methodologies, as well as failure mitigation strategies. However, research on cascade failure prediction is lacking. Wang et al. [25] consider a sequence of failures due to randomly failed physical links (and simultaneous failures of cyber nodes). Makhshari and Mesbah [17] present new research on the challenges of IoT. The authors provide the first systematic study of bugs and challenges that IoT developers face in practice through a large-scale empirical investigation. They collected 5,565 bug reports from 91 representative IoT project repositories and categorized a random sample of 323 based on the observed failures, root causes, and the locations of the faulty components.

Aleš et al. [1], who consider effectiveness of production and maintenance in Industry 4.0 installations with a focus on Industrial IoT, present an interesting approach. In their work, they propose original calculations of Nakajim's OEE (overall equipment effectiveness) indicator for the entire production and monitoring line. This approach, however, relies on relatively simple mathematical models.

As we can conclude, from the observed results, a lot of work was done in the anomaly detection, failure classification and security. The condition of entire networks is a different matter. Efficient diagnostics of large scale IoT device networks are very difficult. Because of device number, maintenance has to be worked in large scale and predictive solutions. Such networks, especially comprising similar devices, can be viewed as populations of sorts, and this is a potential for a new method. The hypothesis that we want to verify in this work is it is possible to use methods usually reserved for populations to analyze networks of IoT devices. We wanted to find out if such methods can provide useful results in the technical context, especially for reliability analysis.

Population behavior is normally a domain of social sciences and psychology. That is why statistical modeling thrives in these areas, with special mention to Bayesian statistics. Bayesian statistics is an approach allowing for inference in the presence of uncertainty (A. Gelman et al. [11]). And in case of analyzing mass-produced products, such as IoT devices, one has to consider production variability and systematic behavior, not unlike 'traditional' populations. This is a field where most successful are multilevel, hierarchical models (Bürkner [6]; Browne and Draper [5]; A. Gelman [12]). There are certain results of Bayesian statistics in the reliability field, as for example Andrzejczak and Bukowski [2] have considered Bayesian approach to model Weibull distribution of expected lifetime. Currently, there is a substantial field of methods allowing practical diagnostics and evaluation of such models (Vehtari, Gelman, and Gabry [24], Gabry et al. [10]; Mikkola et al. [19]). Justification for reliance on population analysis methods also comes from the fact that collection of data regarding IoT behavior is rarely a goal itself, but it's collected as happenstance. So instead of planned validation experiments, we can log information such as restarts, package losses, and times of failure.

The goal of this work is then to provide a conceptual design of a Bayesian method to estimate failure times (or equivalently remaining useful time). Main contributions are:

- creation of data fusion method, joining restart and failure time data, to determine underlying state of system,
- analysis of methods for modeling pooled data to compensate for small variability in the network and their shortcomings when variability increases,
- creation of a multilevel hierarchical model that captures group variability and providing complete design analysis in the aspect of prior selection and computational faithfulness,
- post-stratification with a hierarchical model, using limited data about a sample of population, to predict statistics of the entire group (with potentially different composition) and validating robustness with limited data.

We organized paper as follows. In the following section, we provide basics of Bayesian inference methods used for the rest of the paper. We describe the Bayesian basics, prior and posterior predictive checks, simulation based calibration and computational framework that was used. Then we follow with the description of considered case study. Next three sections cover pooling models for data with similar groups (no faults), multilevel model and post-stratification analysis. Paper ends with conclusions.

2. Materials and Methods

In this section, we will cover the basics of so called Bayesian workflow (Gelman et al. [13]), which is a collection of practices allowing principled and responsible modeling of phenomena. We cover basics and concepts we use in this paper. We also describe the computational setup and express the assumptions of data fusion of observational data for IoT maintenance.

2.1. Bayesian inference

We will now introduce the main ideas of Bayesian modeling, which encapsulate information and data as a highly structured collection of probability distributions. Obviously, such a defined model fully realizes the requirements of transparency and interpretability, as we can instantly recover information about not only feature influence but also their uncertainty.

Principled Bayesian modeling (otherwise known as *Bayesian workflow*) relies on creation of highly customized models that will capture phenomenon of interest. Usually actual process depends on some kind of latent system, and we can observe it through a *measurement process*. Main strength of Bayesian approach is the ability to jointly model this process and the latent system. This is a model of the actual *data generating process*. Our measurement process results in observations $\tilde{y}$, that belong to an *observation space*, Y. On the other hand, models have their assumptions, or structures from the assumptions space $\mathcal{S}$. Parameter θ identify structured models, representing model configurations. Our observational model on the $Y \times \mathcal{S}$ is then the *likelihood*:

$$\pi_{\mathcal{S}}(y \mid \theta) \tag{1}$$

Likelihood together with a *prior distribution* $\pi_{\mathcal{S}}(\theta)$ gives a *Bayesian joint distribution*:

$$\pi_{\mathcal{S}}(y,\theta) = \pi_{\mathcal{S}}(y \mid \theta)\pi_{\mathcal{S}}(\theta) \tag{2}$$

This distribution is a model representing measurement process and the expertise about possible model configurations. Having actual observations we can incorporate them together into *a posterior distribution*:

$$\pi_{\mathcal{S}}(\theta \mid \tilde{y}) = \frac{\pi_{\mathcal{S}}(\tilde{y},\theta)}{\int \pi_{\mathcal{S}}(\tilde{y},\theta)\mathrm{d}\theta} \propto \pi_{\mathcal{S}}(\tilde{y},\theta) \tag{3}$$

which is our main tool for inference.

Important aspect in practice becomes model utility verification. In particular, we need to address prior selection, analysis of prior predictive distribution and simulated based calibration.

2.2. Prior predictive checking

Main tool for prior distribution selection (in a way that allows encoding our expertise) is the prior predictive check. For that, we use the *prior predictive distribution*:

$$\pi_{\mathcal{S}}(y) = \int \pi_{\mathcal{S}}(y,\theta)\mathrm{d}\theta, \tag{4}$$

This distribution allows as to average our data generating process over the prior. This is then a marginal distribution of possible measurements given our expertise. A reasonable prior distribution will provide us the information about observational space expected by the model that it is consistent with observed data. In case of low dimensional problems observation of possible measurement distribution is enough. In multidimensional cases we need to verify distributions of some desired statistics of both simulated and real measurements.

Fortunately, we do not need to construct the distribution (4) explicitly integration might be difficult, impractical or simply not possible. We can, however, approximate it (or pushforward distributions of statistics) using joint distributions samples. Sampling:

$$\begin{aligned} \tilde{\theta} &\sim \quad \pi_{\mathcal{S}}(\theta) \\ \tilde{y} &\sim \pi_{\mathcal{S}}(y \mid \tilde{\theta}) \end{aligned} \tag{5}$$

is equivalent to sampling from the joint distribution (2). In order to get samples from pushforwards we need to evaluate the statistics of interest on values of $\tilde{y}$. We compute Actual prior predictive checks using either visualizations or interval coverages of appropriate marginals.

2.3. Posterior predictive checking

One of the main tools allowing to verify model validity is to verify how predictions generated by the model behave regarding data. Main tool for that is the *posterior predictive distribution*:

$$\pi_{\mathcal{S}}(y \mid \tilde{y}) = \int \pi_{\mathcal{S}}(\theta \mid \tilde{y})\pi_{\mathcal{S}}(y \mid \theta)\mathrm{d}\theta \tag{8}$$

Similarly to the prior predictive distribution, we do not integrate directly, but first sample the posterior distribution of parameters, and then use them to sample the predicted outputs. This is an extremely useful tool, as it allows to compare how the model fits the data and how well it captures the uncertainty.

2.4. Simulated-based calibration

Theoretical tools for model analysis are only worth as much as our ability to compute them. In case of Bayesian models we usually need to use Monte Carlo sampling in some variant (in this paper we use Hamiltonian Monte Carlo). HMC allows for analysis of certain types of problems, like non-identifiability, by analysis of divergences, however there are other issues that make exploration of the posterior (or joint distribution) problematic.

One theoretical tool that we can use for verification of HMC sampling validity is the self consistency of posterior. This property means, that posterior distributions fit using simulated data from prior predictive distribution recover the prior distribution when averaged. We can formulate it as [11]

$$\pi_{\mathcal{S}}(\theta') = \int \pi_{\mathcal{S}}(\theta' | y)\pi_{\mathcal{S}}(y,\theta)\mathrm{d}y\,\mathrm{d}\theta. \tag{7}$$

This is of course needs to be considered in the context of HMC samples. Talts et al. [22] proposed to compare prior distribution and estimated average (7) using rank statistic. If those distributions are equivalent, the distribution of ranks should be uniform. This requires a following procedure. First, we sample parameters from priors. Next, we simulate outputs from corresponding prior predictive distribution, and use them to estimate parameters (in other words, to fit the posterior). We then use prior samples of parameters to compute ranks with respect to sampled parameters from posterior. Because of possibility of autocorrelation influencing rank statistic the samples we thin the samples (usually by a factor of 8). This is repeated many times (usually 1000) and histograms are computed and evaluated for uniformity. Deviation form uniformity indicates computational problems.

2.5. Multilevel regression with post-stratification (MRP)

This method (sometimes called “Mister P”) is a statistical technique used for correcting model estimates for known differences between a sample population (the population of the data you have), and a target population (a population you would like to estimate for). There are multiple success stories of this method. For example, Wang et al. [26] using political surveys for Xbox gamers could construct very reliable predictions of U.S. presidential election results. This idea is popular in social sciences, but as we show here, we can also use it in technical applications.

The post-stratification is essentially using a fitted model for data of a population of different composition than the training dataset. Multilevel regression aspect is representing using Bayesian models for partial pooling of data from different groups within the population. This allows for capturing both similarities and differences between groups.

2.6. Computational setup

For Bayesian computation, we have used Hamiltonian Monte Carlo algorithm, for which the currently most advanced software is Stan (Carpenter et al. [7]). Hamiltonian Monte Carlo algorithm (see, for example, paper by Betancourt [3]) (also known as hybrid Monte Carlo), is a Markov chain Monte Carlo method used for sampling of probability distributions. The main idea behind HMC is to use Hamiltonian dynamics equations to simulate movement of points on the “surface” of probability distribution to construct proposal distribution for Metropolis-Hastings algorithm. Hamiltonian differential equations are automatically created by automatic differentiation of probability distribution and then solved with symplectic integrators (preserving volumes and differential forms, time reversible and limiting diffusion of solutions). The samples from HMC have smaller autocorrelation than, for example, random walk Metropolis-Hastings algorithm and HMC is more efficient in exploration of typical sets of probability distributions. Moreover, HMC has self diagnosis properties as divergences (numerical instabilities in solutions of differential equations) indicate either non-identifiability or problems with posterior geometry.

We performed independent computation on Apple Mac Mini with Apple M1 CPU and 16 GB RAM and on Apple MacBook Pro with Intel Core i9 CPU and 32 GB RAM. Computation results were identical on both platforms, with approximately 14% speed advantage for the former. All the codes are available in the repository listed at the end of the paper.

3. Case study – Diagnostic data fusion

In case of multi-element systems, starting with process industry, through complicated machinery to IoT networks, detailed diagnostics of each part is not possible. One cannot provide sensors for each part, that is why traditional methods of diagnostics are useless. Failure phenomena in all kinds of devices are hard to model, as they usually are tangent to fundamental behavior of systems. It can have multiple, subtle underlying causes. That is why we have two major problems to handle: lack of dedicated data, and no basis in first principle modeling.

That is why we need to consider observational data that can be available with little overhead, in case of IoT devices those are number of resets and total lifetime of device (referred in this paper as failure time). We will create a data fusion [21] of these two disparate sources, in order to provide information about underlying parameters of the model and to predict failure times of devices.

We give the following basic assumptions:

- There are underlying causes of increased number of restarts and approaching failure.
- This relationship is approximately monotonous: high probability of restart reduces expected failure time.
- The approximate model of the underlying cause is a parameter with an appropriate link function corresponding to the state of the system (its health).
- There is a possibility of a variance between batches of the same (or functionally identical) devices, but with some general similarity.
- Distribution of restarts and failure time is Poisson and Gamma, respectively.
- Certain number of restarts are normal.
- Very short failure times are rather unlikely.
- We have an observational data of a few devices, but dataset has representation of this variability.

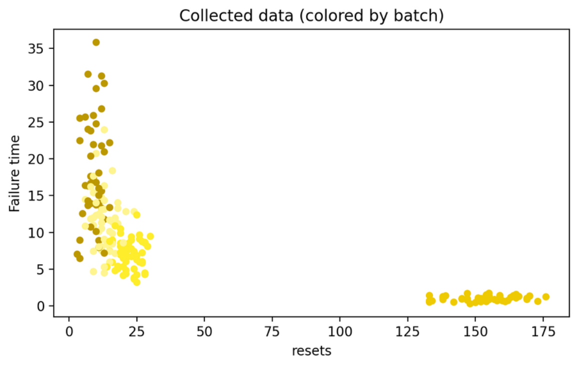

Fig. 1. Relationship between resets occurring during operation and time of failure of the device does not have an obvious relationship. However, we can observe a group of devices that are characterized by a large amount of resets and short lifetime that come from the same batch. Rest of the batches are mixed. Considered dataset comprises 200 samples of pairs of number of resets and registered lifetime

We provide a case study covering an extended network of 1000 similar IoT devices. We have data from 200 devices that have failed; we have registered their failure time and number of resets that occurred. Devices came from four batches approximately evenly represented in the dataset. We artificially generated data, in a way fulfilling the assumptions. Data, together with generating code, are available in the repository. We present registered pairs of resets and failure times in the figure 1. Color of the points corresponds to batches from which they come. As we can observe, three batches express a similar behavior, while one (marked in the dataset as third) exhibits short time till failure and many resets.

In the following sections, we will cover three aspects of modeling:

1. Modeling based only on the healthy data. This is a typical situation, where we have only accessed to normal operation of the system. We will try to cover such a situation with a simple model, pooling all the data together, and then we will analyze how model behaves with addition of a faulty batch.
2. Modeling entire dataset with a full hierarchical model. Because of the increased model complication, we will cover the entire analysis aspect of prior verification and analysis of computational faithfulness.
3. Post-stratification. We will then perform the post-stratification using the hierarchical model on the entire 1000 device network to predict the failure time of the given percentage of the entire network. We will also present the robustness analysis of the model. We will consider how the model behaves when, instead of an original sample of 200 devices, it would consider only its subset. We will verify how it would influence the post-stratification, especially its confidence interval.

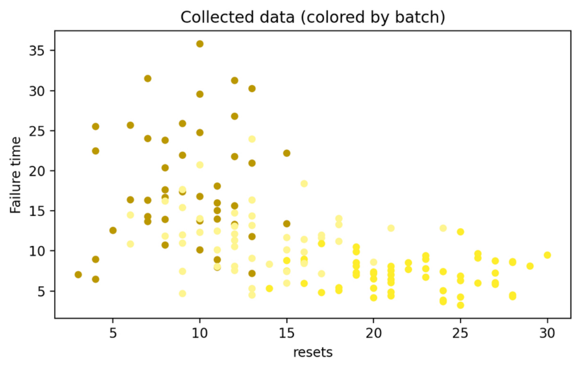

Fig. 2. For the simplified analysis, we consider data from batches numbered as 1, 2 and 4, which exhibit similarity to each other and similar lifetimes. Data includes no more than 30 resets per device

4. Pooling of all data in the normal operation

The first case we want to consider is also a very common one. Often, system data collection is not a planned experiment, but a happenstance, or casual observation. And obviously the most operation occurs under normal conditions, so data about abnormal cases are scarce. That is why we focus on an example of such situation in the considered case. In the figure 2, we see that for three batches behavior of devices is relatively similar, with a bulk of points concentrated in one space.

Because of the observed similarity, we attempt to create a relatively simple model, still adhering to the assumptions. We, however, ignore the information about batches to create a model capturing entire population. This is not an uncommon approach, as not knowing about group variance it is natural to treat all object similar way.

4.1. Poisson model

We propose a three-parameter model with combined likelihood. This likelihood joins both information about resets and the lifetime.

We introduce two additional parameters (besides system health β):

- A reset baseline λ_R which is added to β and through exponential link function serves as a Poisson distribution's rate.
- A shape parameter κ of Gamma distribution, for which (also via link function) β serves as a scale.

Link function is necessary to ensure the positivity of coefficients, while allowing efficient exploration of parameter space by Hamiltonian Monte Carlo.

We present graphical representation of this model in the figure 3, and the mathematical formulas for it are:

$$\begin{aligned} y &\sim \operatorname{Poisson}\left(\exp\left(\lambda_R+\beta\right)\right) \\ \tau &\sim \operatorname{Gamma}\left(\kappa, \exp\left(\beta\right)\right) \\ \kappa &\sim half\,\mathcal{N}\left(8,1\right) \\ \lambda_R &\sim \mathcal{N}\left(3,1\right) \\ \beta &\sim \mathcal{N}\left(0,3\right) \end{aligned} \tag{10}$$

Prior parameters were reasonable guesses to comply with assumptions and not over-saturate the exponential. We do not give here details of prior predictive checks because of space constraints. Code given in the repository allows such an analysis.

We present summaries of samples from the posterior distribution in the table 1. As we can observe, the parameters are relatively well concentrated with λ_R having the smallest standard deviation of all (relative to the mean). Verifying posterior predictive samples in the figure 4 we see, that while for the simple dataset failure times are captured rather well, the predictions of resets have not represented the variance of the sample. This suggests that a more dispersed model can improve it.

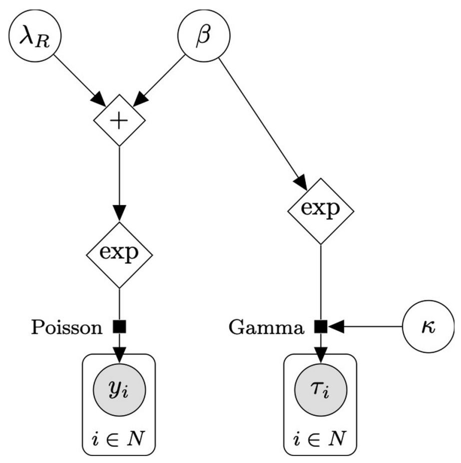

Fig. 3. As an initial attempt of modelling we propose a simple Poisson model, where all devices are characterized by parameter β which is influencing both resets and time of failure. In the proposed model structure we have three parameters. General reset baseline λ_R is added to system parameter and transformed by exponential link function to a Poisson distribution. Link function is necessary to keep positivity of Poisson distribution's rate. Failure time is modelled as a Gamma distribution with shape parameter κ and as the scale we use β, again through link function. For the purpose of image clarity we have denoted number of resets per device as y_i and time of failure of device as τ_i

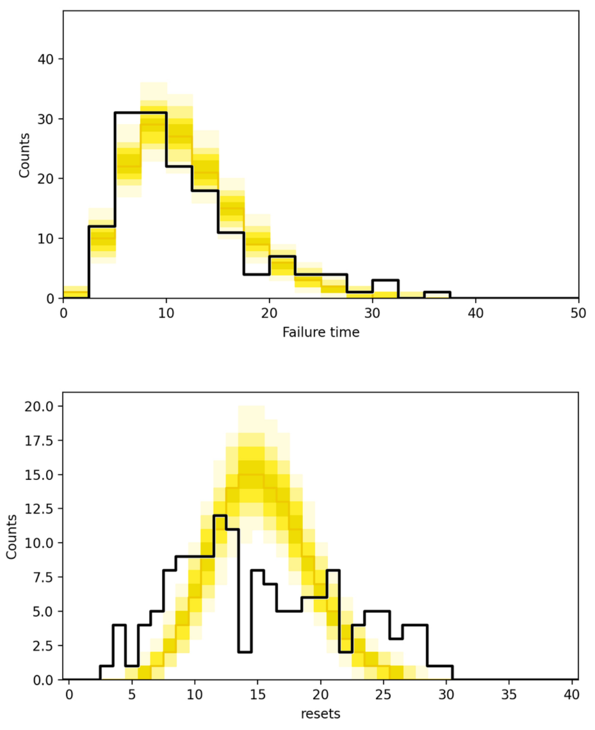

Fig. 4. Posterior predictive distributions of proposed simple model Poisson in reduced dataset are reasonable, but not a perfect fit. We compare ribbon plot of sampled data histograms with the histogram of original data. While the more important failure time (top image) shows reasonable consistency, there is a systematic difference for predictions of number of resets (bottom image). Observed samples are much wider, while predicted ones are well concentrated around the mean

Table 1. Sampling of posterior distribution of the simple model does not provide difficulties. All parameter estimates are well concentrated around their means. Markov Chain Monte Carlo standard errors are small, two orders of magnitude below parameter standard deviations. Effective sample size of both bulk of distribution and its tails is on the level between 16% and 19% (total number of samples is 4000) what is a reasonable result. Potential scale reduction factor , $\hat{R}$ is reasonably close to 1 indicating good mixing of Markov chains

	Mean	st. dev.	$\widehat{\text{MCSE}}$	ESS (bulk)	ESS (tail)	$\hat{R}$
λ_R	3.678	0.104	0.004	652.0	703.0	1.0
κ	4.617	0.441	0.017	666.0	775.0	1.0
β	−0.951	0.102	0.004	647.0	739.0	1.0

mean – mean value of variable, st. dev. – standard deviation of variable, $\widehat{\text{MCSE}}$ – mean of Monte Carlo standard error for variable, ESS (bulk) – effective sample size of samples from the bulk of distribution of variable, ESS (tail) – effective sample size of samples from the bulk of distribution of variable, $\hat{R}$ – potential scale reduction factor

4.2. Dispersed Poisson or Negative Binomial model

One of main limitations of Poisson models (e.g. Poisson regression) is that this distribution enforces mean and variance to be equal. An attractive solution for this requirement is the Negative Binomial distribution, in the so called *location-overdispersion parametrization*[1]:

$$\text{NegBinomial2}(n|\mu,\phi) = \binom{n+\phi-1}{n}\left(\frac{\mu}{\mu+\phi}\right)^n\left(\frac{\phi}{\mu+\phi}\right)^\phi \tag{11}$$

The mean and variance of a random variable $n \sim \text{NegBinomial2}(n \mid \mu,\phi)$ are:

$$\mathbb{E}[n] = \mu \text{ and } \text{Var}[n] = \mu + \frac{\mu^2}{\phi}. \tag{12}$$

Seeing that $\text{Poisson}(\mu)$ has variance μ, negative binomial distribution has its variance increased by an additional $\mu^2/\phi > 0$. This gives much more flexibility in modelling of distributions of integers. The only problem with this parametrization, is that only $\phi \to \infty$ returns to Poisson distribution. That is why it is more convenient to construct model using $\psi = \phi^{-1}$, which actually covers the overdispersion, while $\psi = 0$ returns (computationally) the original Poisson. We will not provide details of the implementation, but the code is in the repository.

Using a negative binomial distribution, we formulate the model in the following form (graphical representation in the figure 5):

$$\begin{aligned} y &\sim \text{NegBinomial2}(\mu,\phi) \\ \mu &= \exp(\lambda_R + \beta) \\ \tau &\sim \text{Gamma}(\kappa,\exp(\beta)) \\ \kappa &\sim \mathit{half}\,\mathcal{N}(8,1) \\ \lambda_R &\sim \mathcal{N}(3,1) \\ \beta &\sim \mathcal{N}(0,3) \end{aligned} \tag{13}$$

Regarding prior on ϕ , we actually set uniform, bounded from below by 0 prior on $\psi = \phi^{-1}$ This essential gives a reciprocal of square for prior on ϕ , but it does not influence computation, and would only clutter the formulas. In the code, we represented it by setting a lower bound of 0 on the parameter.

We present summaries of samples from the posterior distribution in the table 2. All the parameters are much more uncertain than in the previous case, which actually is a desired quality. As we know, data came from different batches, and capturing variance between them requires more uncertain parameters. β parameter has even unknown sign (as 67% confidence interval has zero somewhere around the middle), but this is not as important because of link function. This uncertainty, however, impedes the predictive quality of failure time. As we can see in the figure 6, the ribbon plot of failure time predictions is much wider, and the actual median prediction is well below observed data. Ability to capture resets variance in the right image in figure 6 results in unreliable prediction for failure time in the left image. Most uncertainty is especially for short failure times.

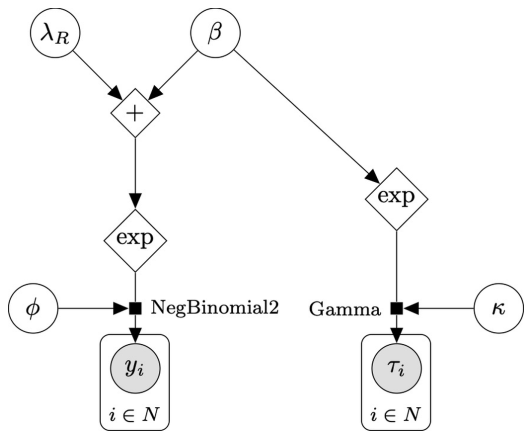

Fig. 5. In order to better capture the variance in device resets in the case of similar groups we are considering the overdispersed integer model, governed by the negative binomial distribution. The general assumptions are not modified – all devices are characterized by parameter β which is influencing both resets and time of failure. Similarly to the poisson model, we are adding the exponential link and a reset baseline λ_R. We are adding however the over-dispersion parameter which functionally behaves like poisson distribution, but with variance different from the mean. Failure time, without changes, is modelled as a Gamma distribution with shape parameter κ and as the scale we use β, again through link function. For the purpose of image clarity, we have denoted number of resets per device as y_i and time of failure of device as τ_i

As we can observe, both models pooling all the data together have their shortcomings. For completeness, we present how they predicted resets and failure times for individual batches. We can observe it in the figure 7. As we can see, predictions for the population are not representative of the individual groups. This puts into question how well both models would function in the full dataset.

1 see for example Stan Functions Reference https://mc-stan.org/docs/2_28/functions-reference/nbalt.html

Table 2. Sampling of the posterior distribution of the negative binomial model is computationally more efficient. Effective sample sizes start at 32% of original sample size, even getting over 50% in one case. $\hat{R}$ is very close to 1. However, uncertainty of parameters is much more significant, up to the situation that we cannot confidently establish the sign of β. Markov Chain Monte Carlo standard errors are also larger, but this is explainable by increased variance. This is justified because data comes from different distributions, and some compromises had to be made

	Mean	st. dev.	$\widehat{\text{MCSE}}$	ESS (bulk)	ESS (tail)	$\hat{R}$
λ_R	2.957	0.925	0.026	1302.0	1448.0	1.0
κ	7.983	0.967	0.020	2303.0	1910.0	1.0
β	-0.230	0.926	0.026	1301.0	1474.0	1.0
ϕ	7.632	1.367	0.028	2438.0	2076.0	1.0

mean – mean value of variable, st. dev. – standard deviation of variable, $\widehat{\text{MCSE}}$ – mean of Monte Carlo standard error for variable, ESS (bulk) – effective sample size of samples from the bulk of distribution of variable, ESS (tail) – effective sample size of samples from the bulk of distribution of variable, $\hat{R}$ – potential scale reduction factor

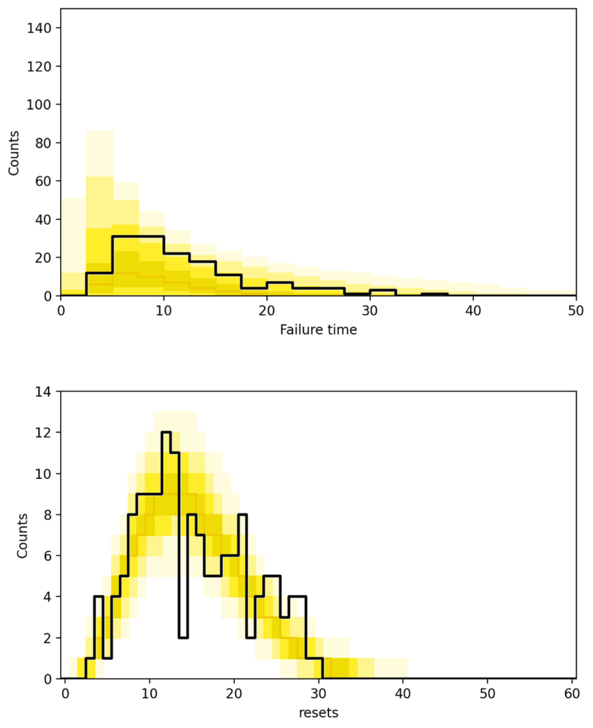

Fig. 6. Posterior predictive distributions of proposed dispersed model improve the predictive ability in the aspect of resets (bottom image) but the increased uncertainty of parameters results in very wide confidence intervals for short failure time occurrences (top image)

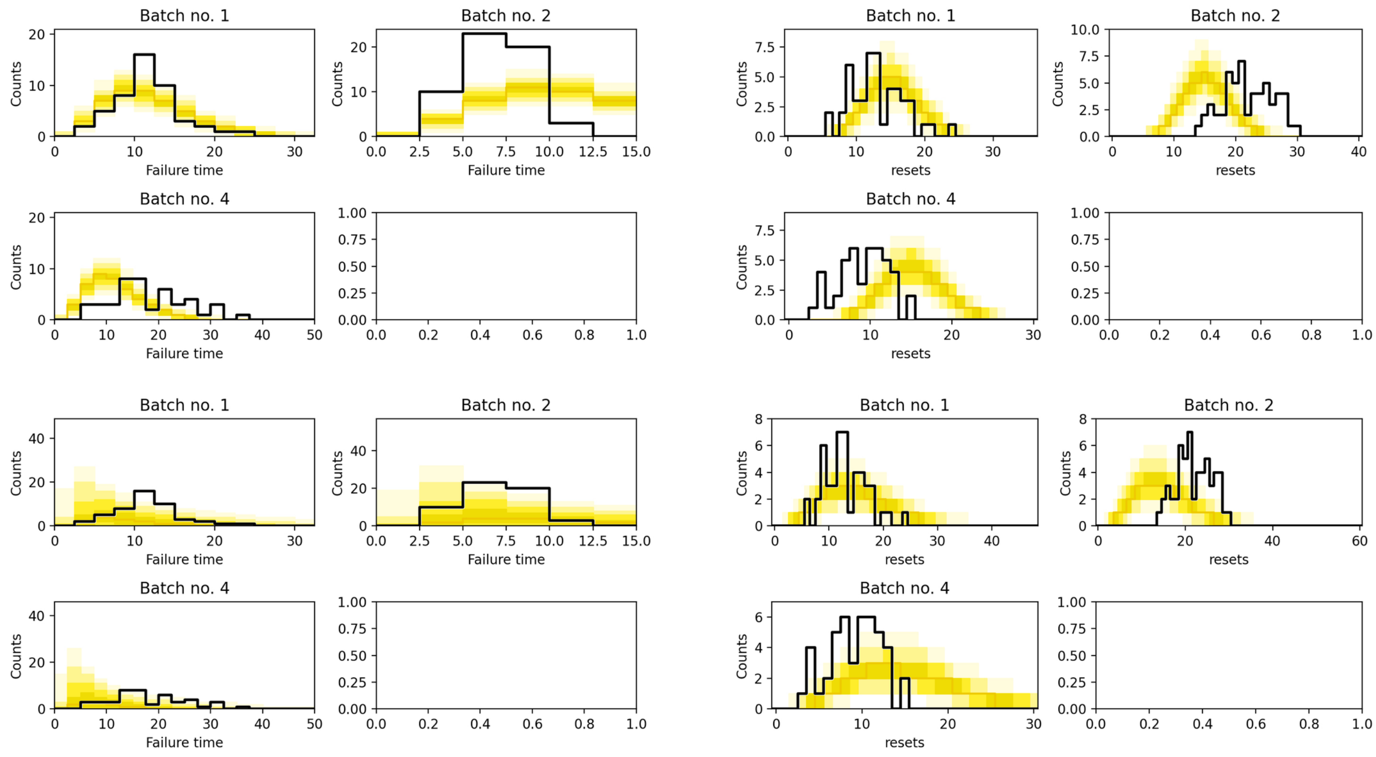

Fig. 7. Posterior predictive distributions of Poisson (top two rows) and negative binomial (bottom two rows) show that neither model using pooled data can capture behavior within individual groups. Batch 3 is missing, as it contains the faulty data that we will introduce to the model in the next section

4.3. Inference on full dataset

While simple models worked rather well evaluating pooled data for similar groups, introduction of the less similar group (batch, see Fig. 1) show their weakness more significantly. In order to save space, we will not provide here full results of sampling (they are available in the repository), but we just give some main takeaways:

- Sampling of both models was rather similar regarding parameter concentration and HMC behavior.
- We observed that λ_R increased significantly for the Poisson model (mean of 5).
- While both models incorporated the same distribution for failure time, its coefficients were completely different. For this case β

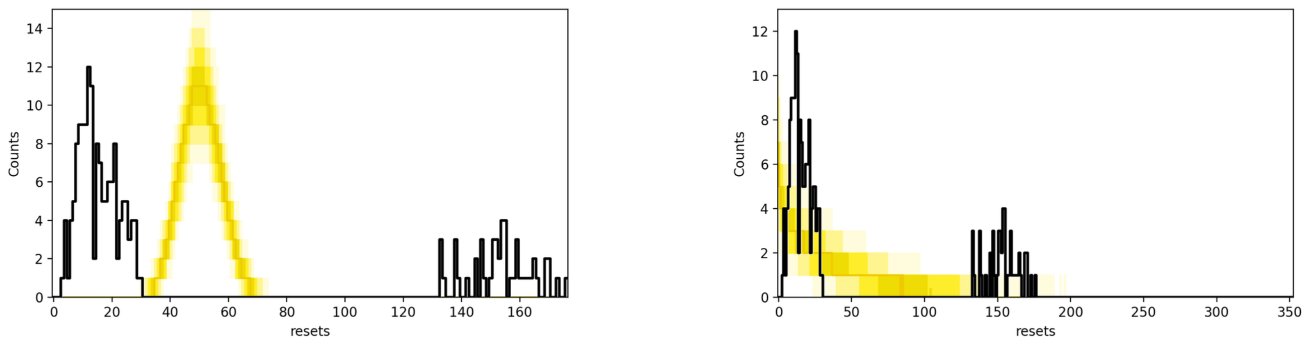

Fig. 8. Posterior predictive distributions of resets for Poisson (left image) and negative binomial (right image) models show that neither model can handle introduction of more different groups. Poisson distribution just give prediction between the observed data, and the negative binomial model is over-inflated at zero

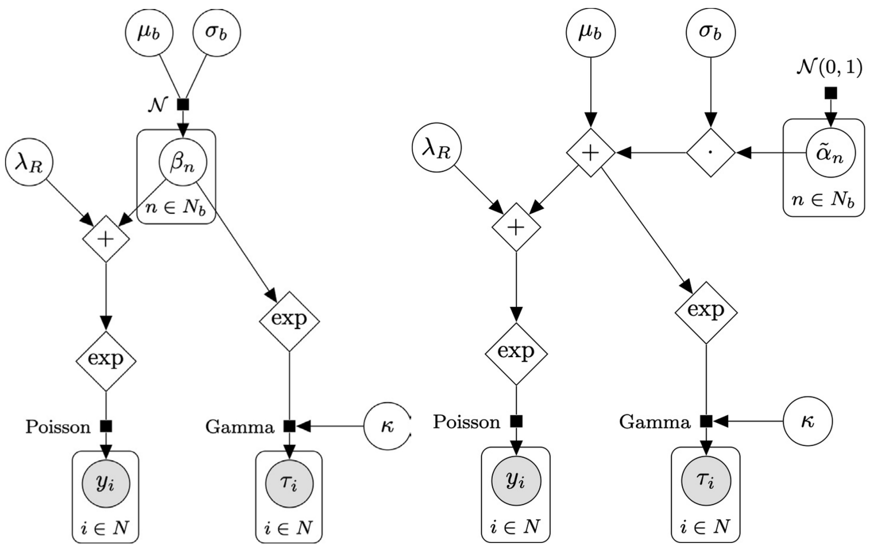

Fig. 9. In order to capture between group behavior, we used a multilevel regression model with use of batch indicator as predictor with individual intercepts for each group. Because we are considering the same type of device expected behavior is similar, that is why the model is hierarchical with normal prior for intercepts with hyper-parameters μ_b and σ_b. Natural, so called 'centered' parametrization (left image) is, however, very inefficient computationally. We do not provide detailed analysis here. Those problems are not present in equivalent 'non-centered' parametrization (right image)

was strongly negative for Poisson model, and mostly positive for the negative binomial one.

- Neither model could capture behavior of resets. In the figure 8, we can see that Poisson model is stuck between actual data and negative binomial is very vague.
- We are not presenting failure time predictions, nor individual group predictions, as they are rather uninteresting. Complete set of figures with generating code is available in the repository.

These issues show that construction of a model cannot ignore group behavior, especially if we expect groups to differ from one another. In case of IoT devices, changes in manufacturer, model or even production batch can introduce significant deviations. That is why we need to focus on more sophisticated models.

5. Focusing on individual batches – Multilevel hierarchical model

We propose to use multilevel hierarchical model. Batches will share the same distribution type, but with different parameters, coming from a common distribution with unknown hyper-parameters. We give proposed model structure in the figure 9 in both centered and non-centered parametrization. In our code, we only use the latter because of its computational efficiency. We avoided providing detailed equations, because they are more readable from the code or diagram than from equations.

Main justifications of the model are following:

- We assume general structure of Poisson-Gamma model for each group.
- We keep reset baseline λ_R and shape κ identical for each group.
- Multilevel effect come from the individual 'health' parameter β_i for each group (batch) of devices.
- Hierarchy is in the regularization enforced by the assumption that all β_i come from the same Normal distribution with hyper-parameters μ_b and σ_b. Hyper-parameters have respectively priors as standard normal, and half-normal with standard deviation of 2.
- Non-centered parametrization is functionally equivalent, but is computationally more efficient to consider group parameters from standard normal and scaling and shifting them by parameters.

5.1. Prior predictive checking

Because the model is much more complicated, we provided a thorough prior predictive checks illustrated in table 3 and with prior predictive distributions in the figure 10. The general idea is to sample parameter values from priors. So using priors we have simulated 1000 times parameters for 200 devices split into 4 batches (with the same composition of batches as in original data) and then using those parameters sampled from appropriate Poisson and Gamma distributions corresponding pairs of resets and failures. Having such samples, we have computed relevant statistics. In table 3, we can see that statistics of individual parameters are consistent with priors. We visualize simulated data in the figure 10. We have used the same bins as with histogram of original data and computed histograms for each sample (of 200 devices), then for each bin we have computed quantiles obtaining the visualized ribbon plots. Data simulated on priors is obviously zero inflated, but it does not contradict the original data, as it fits within the ribbon plots.

5.2. Simulation-based calibration

To ensure computational fidelity, we have performed full simulation-based calibration. We did this to avoid situation, when complicated model geometry provides difficulties in proper exploration of typical set by HMC in Stan, and also to avoid potential coding errors. Using a similar scheme as with prior predictive checks, we have simulated 1000 samples of 200 devices and their corresponding resets and failure times. For each of those samples, we have performed MCMC sampling of the multilevel model using them as data. Then, for each of sampled parameters, we have computed the rank statistic, of how

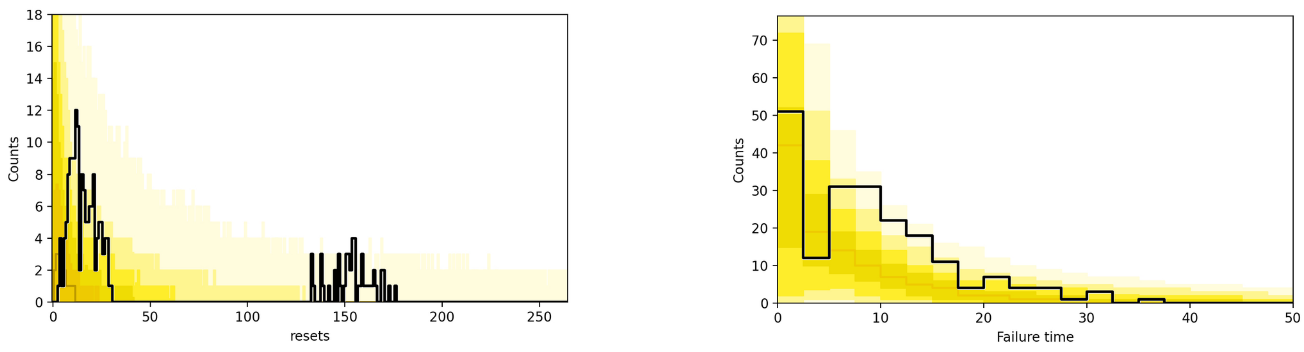

Fig. 10. Complicated structure of the model warrants analysis of parameter priors, so they allow consistency with data. Provided prior parameters show that failure time is plausible (left image). Reset data is not as clear, as histogram does not fit in the 90th percentile of ribbon plot. It, however, fits in to the 99th (right image)

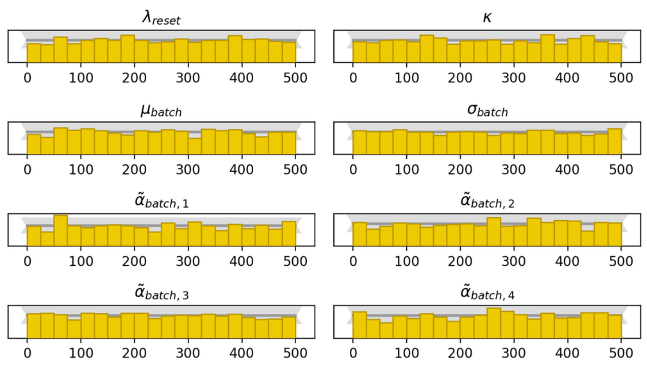

Fig. 11. In order to verify the computational faithfulness of the model, we have performed simulation-based calibration. We have sampled from prior predictive distribution and fitted the model to those samples. Because of self-consistency property of Bayesian models, the rank statistic of parameters should have uniform distribution. Grey shapes correspond to percentiles of acceptability (with the dark gray line being median). As we can see it, all rank histograms are reasonably uniform

many samples were smaller than the one of the model which generated the data. Distributions of those ranks should be uniform. We verified that by thinning samples 8 times and plotting histograms. We present histograms of ranks in the figure 11. All are reasonably uniform.

5.3. Posterior predictive (retrodictive) distributions

We complete our analysis with posterior predictive checks, which work as an actual validation of the model. As previously, we can verify that using ribbon plots and comparing them to the data. In this way we can consider them retrodictive, as we want to obtain the same data, that we used for the inference.

In the figure 12, we can see that, especially in the aspect of resets, we have obtained very good results. While median of predicted failure times is consistent with data, previous models also were not bad in this aspect. However, in case of resets we can see, that model well identified groups of devices. This is also visible in plots for the individual groups (figure 13) as the model coefficients well represented each batch.

Finally, we can observe the parameter fits in table 4. We can see the relatively good concentration of parameter values for each group β_i. What is perhaps interesting, such concentration is visible in 'centered' parameters and not in the 'non-centered' that are the part of the actual model. Hyper-parameters have significant uncertainty, allowing all the individual batches to be captured. As we can also see, batch no. 3, the faulty one, has significantly different coefficient.

6. Post-stratification and robustness analysis

Having a well-fitted model it was possible to perform post-stratification. We have tested it on a sample of 1000 devices with only their batch number recorded. Our goal was to analyze following statistics:

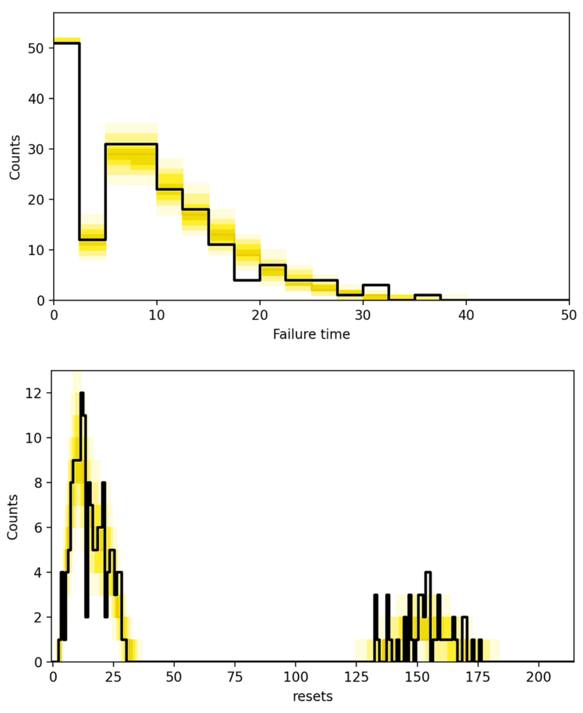

Fig. 12. Application of multilevel model results in much better fits to the data. Histogram of failure times is almost identical to ribbon plot median (top image). In case of resets, our histogram is independent for every integer, so we can see certain negligible inconsistencies (bottom image)

- after what time 20% devices will fail,
- after what time 50% devices will fail.

The idea of post-stratification computationally uses the samples from the MCMC inference. In our case, we had 4000 samples of parameters for each group and the global parameters. For each of those, we sampled 1000 failure times (with the desired composition of devices) and computed the relevant statistic. In this way, we obtained 4000 samples of each statistic, which we could use for our maintenance predictions. We visualize these results in the figure 14. As we can see, their distribution is approximately normal and value concentrations are relatively good. As we can see in the table 5, confidence intervals are tight with standard deviations on the level of 4% of mean.

6.1. Robustness to amount of data

As a final verification of the methodology, we have analyzed the model consistency if scarcer data are available. Considered 200 samples is a relatively big part of post-stratified 1000, and not I realistic.

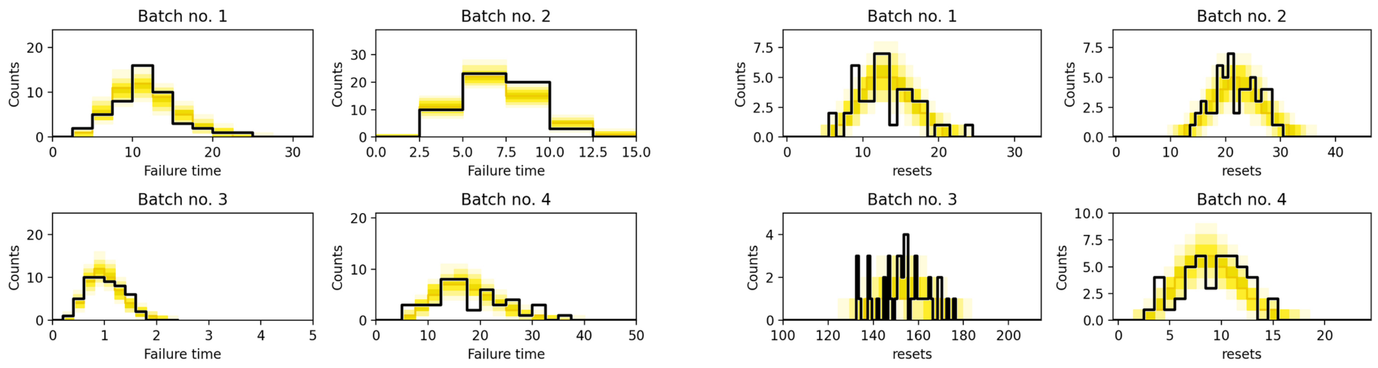

Fig. 13. Analysis of the multilevel model with respect to individual batches supports it is a good model for the problem. Each individual group is well modeled, both for cases of failure times (a) and resets (b). In case of batch 3 i.e. the faulty one, the fit of resets is less obvious, but we should note that all the samples are in the neighborhood of data points, and the actual number of datapoints observed was not large for each integer value

Table 4. Posterior parameter estimates are reasonable. Each of them has a confidence interval with a sensible level of uncertainty. Sampling is very effective as effective sample sizes are around 25% some even exceeding. For completeness, we present also recomputed parameters β which are not explicitly present in non-centered parametrization. From it, we can clearly see that parameter of batch no. 3 differs significantly from the others. Potential scale reduction factor $\hat{R}$ is reasonably close to 1 indicating good mixing of Markov chains, however it was required to adapt integration step of HMC and increase the maximal tree depth parameter.

	Mean	st. dev.	$\widehat{\text{MCSE}}$	ESS (bulk)	ESS (tail)	$\hat{R}$
λ_R	2.960	0.080	0.002	1337.0	1671.0	1.00
κ	8.164	0.615	0.017	1262.0	1378.0	1.00
μ_{batch}	0.145	0.621	0.020	988.0	1170.0	1.00
σ_{batch}	1.497	0.609	0.019	990.0	1248.0	1.00
$\tilde{\alpha}_{batch,1}$	−0.411	0.453	0.016	813.0	1083.0	1.00
$\tilde{\alpha}_{batch,2}$	−0.019	0.420	0.014	893.0	1035.0	1.01
$\tilde{\alpha}_{batch,3}$	1.462	0.646	0.020	1079.0	1121.0	1.00
$\tilde{\alpha}_{batch,4}$	−0.712	0.504	0.018	773.0	980.0	1.00
β_1	−0.376	0.084	0.002	1412.0	1596.0	1.00
β_2	0.137	0.082	0.002	1398.0	1781.0	1.00
β_3	2.075	0.080	0.002	1349.0	1715.0	1.00
β_4	−0.770	0.083	0.002	1479.0	1922.0	1.00

mean – mean value of variable, st. dev. – standard deviation of variable, $\widehat{\text{MCSE}}$ – mean of Monte Carlo standard error for variable, ESS (bulk) – effective sample size of samples from the bulk of distribution of variable, ESS (tail) – effective sample size of samples from the bulk of distribution of variable, $\hat{R}$ – potential scale reduction factor

Recent works (Broderick, Giordano, and Meager 2021) show that even 1% of data can significantly influence the predictions. While authors provide more detailed metrics for certain simpler problems, we have dropped randomly bigger chunks of data to see how it influences the post-stratification results. We have decided to randomly drop 10%, 75% and even 90% of data to see how post-stratification results will behave. As a metric, we have considered means and standard deviations of statistics of interest. We summarize results in the figure 15 and the table 6. As we can see, the method handles issue remarkably well. Means are very consistent and what really changes are the confidence intervals, which also are not blowing up, as they are at most doubled (better than the expected $\sqrt{10}$). What is also interesting (but because of constraints kept to the repository) is that even with 20 data points, the model still captures individual properties of each group rather well (and estimate 8 parameters).

7. Discussion and conclusions

In this paper, we have performed a conceptual analysis of using Bayesian models for modeling IoT device behavior using Bayesian analysis. Our intention was to show that such models allow capturing complicated group behavior (for different batches of devices). We have shown, that for simpler cases, with limited variation between units we can ignore it all together with obtaining estimates with some

Table 5. Results of post stratification in tabular form. Q_{20} is the estimated 20th percentile of failure times, and median is its median. Estimates are relatively tight

	Mean	st. dev.	$\widehat{\text{MCSE}}$	ESS (bulk)	ESS (tail)	$\hat{R}$
q_{20}	5.971	0.245	0.004	3501.0	3559.0	1.0
median	10.064	0.362	0.006	3876.0	3530.0	1.0

mean – mean value of variable, st. dev. – standard deviation of variable, $\widehat{\text{MCSE}}$ – mean of Monte Carlo standard error for variable, ESS (bulk) – effective sample size of samples from the bulk of distribution of variable, ESS (tail) – effective sample size of samples from the bulk of distribution of variable, $\hat{R}$ – potential scale reduction factor

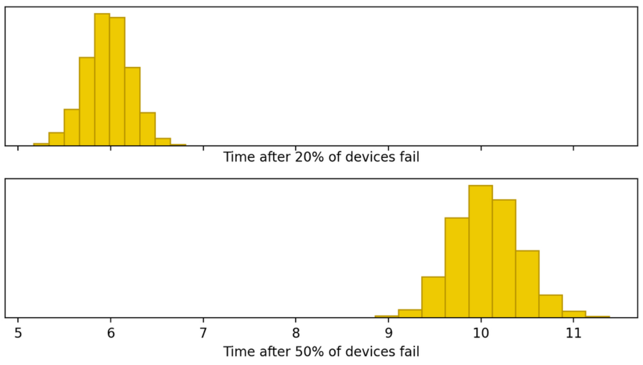

Fig. 14. Using model estimated using 200 samples with approximately even batch distribution, we can compute quantities of interest. In particular, we can estimate the distribution of time after 20% of devices fail (top plot) and same for 50% (bottom plot)

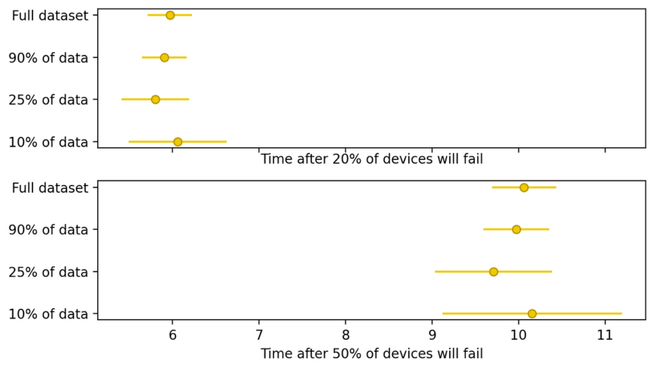

Fig. 15. Reducing the dataset does not influence the quality of post-stratification significantly. While having fewer samples increases the confidence interval, mean is relatively consistent and loss of uncertainty is not enormous. In the image, we have visualized means along with error bars of one standard deviation

uncertainty. When the device population has large variation (for example from bad quality control at the manufacturer or subcontractor) it justifies more complicated models. We have shown that using hierarchical multilevel structure, we can efficiently estimate population lifetime. Proposed model is robust with respect to data reduction, provided groups are representation. These results show great potential for using Bayesian method in technical applications, where they are underrepresented, especially for reliability analysis.

There are obvious limitation to our approach. Because we have used simulated data, it is obviously not applicable directly. However, using real data would obfuscate main ideas, as it would necessitate more complex model. And such complications are a potential for method extension.

Table 6. Detailed values of estimated q_{20} (the estimated 20th percentile) and median of failure times, for different amounts of data. Mean values are very consistent, and data reduction (even 10 times) increases uncertainty in a limited fashion

		mean	st. dev.
10% of data	q_{20}	6.059	0.555
	median	10.156	1.025
25% of data	q_{20}	5.803	0.379
	median	9.710	0.665
90% of data	q_{20}	5.907	0.248
	median	9.973	0.370
Full dataset	q_{20}	5.971	0.245
	median	10.064	0.362

Natural ideas of extension would model not only within device type but also between type variance of devices. As long as we can capture certain similarities, concepts still hold. This conceptually is just adding a level to the model. We have just used examples of post-stratification metrics. Those could be much more advanced utility functions. For example, one could consider expected coverage, energy losses, costs of maintenance, etc. Such complications of the model are possible and we will investigate them in the future.

The observation about robustness is an assuring one, as 20 datapoints allowed to estimate 8 parameters of the model with accuracy high enough to obtain useful results. Reason of such gain might be the data fusion effect. Using two measurements from each device adds additional regularization. This compensated for an increased number of parameters.

There is a large application potential for the depth of Bayesian methods in the technical sciences. This work is one example and we will continue it in further research. Monitoring and maintaining IoT networks is one area where their use can be very natural.

Acknowledgments

This research was funded by AGH's Research University Excellence Initiative under project "Interpretable methods of process diagnosis using statistics and machine learning", by AGH's subvention for scientific activity and by Polish National Science Centre project "Process Fault Prediction and Detection" contract no. UMO-2021-/41/B/ST7/03851

The data presented in this study and code for the analysis are openly available in the GitHub repository https://github.com/KAIR-ISZ/public_data.

Author would like to thank Prof. Edyta Kucharska and dr Katarzyna Grobler-Debska for stimulating discussions

Abbreviations

We use the following abbreviations in this manuscript:

HMC	Hamiltonian Monte Carlo
HDI	Highest Density Interval
ESS	Effective Sample Size
MCSE	Monte Carlo Standard Error

References

1. Aleš Z, Pavlů J, Legát V et al. Methodology of overall equipment effectiveness calculation in the context of Industry 4.0 environment. Eksploatacja i Niezawodnosc - Maintenance and Reliability 2019; 21(3): 411–418, https://doi.org/10.17531/ein.2019.3.7.
2. Andrzejczak K, Bukowski L. A method for estimating the probability distribution of the lifetime for new technical equipment based on expert judgement. Eksploatacja i Niezawodnosc - Maintenance and Reliability 2021; 23(4): 757–769, https://doi.org/10.17531/ein.2021.4.18.
3. Betancourt M. A Conceptual Introduction to Hamiltonian Monte Carlo. arXiv:1701.02434 [stat] 2018.
4. Broderick T, Giordano R, Meager R. An Automatic Finite-Sample Robustness Metric: When Can Dropping a Little Data Make a Big Difference? arXiv:2011.14999 [econ, stat] 2021.
5. Browne W J, Draper D. A comparison of Bayesian and likelihood-based methods for fitting multilevel models. Bayesian Analysis 2006, https://doi.org/10.1214/06-BA117.
6. Bürkner P-C. Advanced Bayesian Multilevel Modeling with the R Package brms. The R Journal 2018; 10(1): 395, https://doi.org/10.32614/RJ-2018-017.
7. Carpenter B, Gelman A, Hoffman M D et al. Stan : A Probabilistic Programming Language. Journal of Statistical Software 2017, https://doi.org/10.18637/jss.v076.i01.
8. Diro A A, Chilamkurti N. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems 2018; 82: 761–768, https://doi.org/10.1016/j.future.2017.08.043.
9. Fahim M, Sillitti A. Anomaly Detection, Analysis and Prediction Techniques in IoT Environment: A Systematic Literature Review. IEEE Access 2019; 7: 81664–81681, https://doi.org/10.1109/ACCESS.2019.2921912.
10. Gabry J, Simpson D, Vehtari A et al. Visualization in Bayesian workflow. Journal of the Royal Statistical Society: Series A (Statistics in Society) 2019; 182(2): 389–402, https://doi.org/10.1111/rssa.12378.
11. Gelman A, Carlin J B, Stern H S et al. Bayesian Data Analysis, Third Edition. Taylor & Francis: 2013.
12. Gelman A. Parameterization and Bayesian Modeling. Journal of the American Statistical Association 2004; 99(466): 537–545, https://doi.org/10.1198/016214504000000458.
13. Gelman A, Vehtari A, Simpson D et al. Bayesian Workflow. arXiv:2011.01808 [stat] 2020.
14. Hasan M, Islam Md M, Zarif M I I, Hashem M M A. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet of Things 2019; 7: 100059, https://doi.org/10.1016/j.iot.2019.100059.
15. Kaya A, Keçeli A S, Catal C, Tekinerdogan B. Sensor Failure Tolerable Machine Learning-Based Food Quality Prediction Model. Sensors 2020; 20(11): 3173, https://doi.org/10.3390/s20113173.
16. Lin Y-B, Lin Y-W, Lin J-Y, Hung H-N. SensorTalk: An IoT Device Failure Detection and Calibration Mechanism for Smart Farming. Sensors 2019; 19(21): 4788, https://doi.org/10.3390/s19214788.
17. Makhshari A, Mesbah A. IoT Bugs and Development Challenges. 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), Madrid, ES, IEEE: 2021: 460–472, https://doi.org/10.1109/ICSE43902.2021.00051.
18. Manimurugan S, Al-Mutairi S, Aborokbah M M et al. Effective Attack Detection in Internet of Medical Things Smart Environment Using a Deep Belief Neural Network. IEEE Access 2020; 8: 77396–77404, https://doi.org/10.1109/ACCESS.2020.2986013.
19. Mikkola P, Martin O A, Chandramouli S et al. Prior knowledge elicitation: The past, present, and future. arXiv:2112.01380 [stat] 2021.
20. Moghaddam M T, Muccini H. Fault-Tolerant IoT: A Systematic Mapping Study. In Calinescu R, Di Giandomenico F (eds): Software Engineering for Resilient Systems, Cham, Springer International Publishing: 2019; 11732: 67–84, https://doi.org/10.1007/978-3-030-30856-8_5.
21. Stief A, Ottewill J R, Baranowski J, Orkisz M. A PCA and Two-Stage Bayesian Sensor Fusion Approach for Diagnosing Electrical and Mechanical Faults in Induction Motors. IEEE Transactions on Industrial Electronics 2019; 66(12): 9510–9520, https://doi.org/10.1109/TIE.2019.2891453.
22. Talts S, Betancourt M, Simpson D et al. Validating Bayesian Inference Algorithms with Simulation-Based Calibration. arXiv:1804.06788 [stat] 2020.
23. Vangipuram R, Gunupudi R K, Puligadda V K, Vinjamuri J. A machine learning approach for imputation and anomaly detection in IoT environment. Expert Systems 2020, https://doi.org/10.1111/exsy.12556.
24. Vehtari A, Gelman A, Gabry J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Statistics and Computing 2017; 27(5): 1413–1432, https://doi.org/10.1007/s11222-016-9696-4.
25. Wang J, Pambudi S, Wang W, Song M. Resilience of IoT Systems Against Edge-Induced Cascade-of-Failures: A Networking Perspective. IEEE Internet of Things Journal 2019; 6(4): 6952–6963, https://doi.org/10.1109/JIOT.2019.2913140.
26. Wang W, Rothschild D, Goel S, Gelman A. Forecasting elections with non-representative polls. International Journal of Forecasting 2015; 31(3): 980–991, https://doi.org/10.1016/j.ijforecast.2014.06.001.
27. Xing L. Cascading Failures in Internet of Things: Review and Perspectives on Reliability and Resilience. IEEE Internet of Things Journal 2021; 8(1): 44–64, https://doi.org/10.1109/JIOT.2020.3018687.