Lifetime Predictions of Electrolytic Capacitors in Network Cameras with Random Forest

Oscar Andersson and Oskar Hindgren

LUND UNIVERSITY

Department of Automatic Control & Centre for Mathematical Sciences

MSc Thesis
TFRT-6114
ISSN 0280-5316

Department of Automatic Control & Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2020

Acknowledgements

Firstly, we wish to thank our supervisor Jonna Stålring Westerberg, for all of her time, valuable insights and guidance. We consider ourselves very fortunate for having such a knowledgeable and encouraging supervisor.

Secondly, Anton Friberg also deserves a large thank you for always giving us a helping hand whenever we were confronted with problems concerning either Linux or Python.

Furthermore, Thomas Vallier is entitled to be mentioned and thanked, as he helped to develop the exceptional Vallier interpolation method. And also, for spending time proofreading the paper on multiple occasions.

In addition, we wish to thank, our other colleagues at Axis for all their help, support, and for taking an interest in our work, Anja Lemic, Jon Hansson, Nicklas Hörnqvist, Ola Söder, Pontus Rosenberg, and Tory Li.

We also wish to thank our supervisors at the university, Bo Bernhardsson, and Kalle Åström, for their time and thoughtful feedback on our work.

Lastly, we would like to thank our loving families and encouraging friends for supporting us throughout this work.

Abstract

Electrolytic capacitor components degrade when exposed to thermal stress which can cause failures in electrical devices. Several recent works have studied the lifetime of these components by using accelerated life testing. This work, however, takes a new approach by utilising vast amounts of temperature data and regression techniques. Axis Communications collects large amounts of non-personalised data from network cameras in real-time, which could be used for lifetime predictions. However, there are various problems with the collected data, such as jitter, interruptions, and missing data. Methods to resolve these problems are developed and validated.

To predict the lifetime of an individual two different models are developed, a Random forest model and a Baseline model. The Baseline model is used as a validation of the performance of the Random forest model. The models require temperature data to create predictions. The goal is to achieve a mean absolute normalised error of less than 10 %, while simultaneously minimising the required amount of data. The Random forest model achieves the target mean absolute normalised error with significantly less data than the Baseline model.

Furthermore, distributions of the lifetime predictions are formed, as they could help guide future product design. The distribution of the predictions is compared to the true distribution with statistical tests. The distribution of the Random forest predictions is concluded to be more similar to the true distribution than the Baseline model.

Keywords: *Random forest, Lifetime predictions, Electrolytic capacitors, Thermal stress, Network cameras, Missing value imputation.*

Abbreviations

AJ	**A**rtifical **J**itter
BM	**B**aseline **M**odel
CDF	**C**umulative **D**istribution **F**unction
CV	**C**ross **V**alidation
DTB	**D**ata **T**ransfer **B**ox
EDF	**E**mperical **D**istribution **F**unction
i.i.d.	**I**ndependent and **I**dentically **D**istributed
IoT	**I**nternet **o**f **T**hings
KS	**K**olmogorov-**S**mirnov
L	**L**ifetime
LC	**L**ifetime **C**onsumption
MANE	**M**ean **A**bsolute **N**ormalised **E**rror
MCAR	**M**issing **C**ompletely **A**t **R**andom
ML	**M**achine **L**earning
MSE	**M**ean **S**quared **E**rror
RF	**R**andom **F**orest
TS	**T**ime **S**eries
YLC	**Y**early **L**ifetime **C**onsumption
YTL	**Y**early **T**ime to **L**ive

Contents

1

Introduction

1.1 Background

The number of internet of things (IoT) devices is growing, resulting in an increased demand for maintenance and opportunities for continuous and automated device health monitoring. Forecasting the health of devices is a growing application of machine learning (ML). The forecasts are important tools for optimising the use of resources and to increase customer satisfaction. ML applications require data. Axis Communications (Axis) has extensive time series (TS) data (non-personal) on network cameras and several metrics that might be indicative of the device health of a camera individual. Such as image quality, process-specific memory and CPU consumption, as well as the temperature and power consumption of various components. This thesis will focus on modelling the thermal stress in electrolytic capacitors by studying temperature time series.

Electrolytic capacitors are polarised capacitors composed of an electrolyte-impregnated paper layer sandwiched between two highly roughened metal foils (usually aluminium) serving as anode and cathode [Gupta et al., 2018]. The electrolytic capacitors provide high capacitance values, high volumetric efficiency, and an excellent price over performance ratio [Gupta et al., 2018].

However, the electrolytic capacitor generally has the shortest lifespans among the components in power electronics [Gupta et al., 2018]. These capacitors are present in most Axis camera products, and engineers at Axis consider them to be amongst the most temperature sensitive components in the cameras. Consequently, it is of great importance to study the lifetime and reliability of these components. The constructed models of lifetime are essential for designers to design and guarantee the reliability/life of electrical components.

One of the primary wear-out mechanisms in electrolytic capacitors is the loss of electrolyte by vapour diffusion through the seals [Sankaran et al., 1997]. Other reports suggest that the main wear out mechanism is deterioration of the electrolyte. This causes the capacitance to degenerate and when the capacitance decreases be-

low a particular value (often 20 percent of its initial value), the capacitor will no longer be able to store the necessary energy [Cherry et al., 2018], which causes other components to malfunction.

Generally, capacitors are studied by accelerated life testing techniques [Cherry et al., 2018], [Albertsen, 2010], where the capacitors are exposed to more harsh conditions compared to their normal operating condition. However, this report proposes a novel approach by studying actual temperature data measured from the source.

The expected lifetime of a capacitor can be quantified from a formula depending on temperatures [Cherry et al., 2018]. Provided that a temperature time series is available for a year, it is possible to determine how much of a capacitor's lifetime is consumed during that year. From a health monitoring point of view, it would be useful to be able to predict the yearly lifetime consumption (YLC) of heat tolerance for a camera knowing the temperature time series for less than a year. Therefore, studying how the YLC could be predicted and the amount of data required to obtain reasonable prediction accuracy would be of value to Axis. Furthermore, once YLC is available for a larger population of cameras, it can guide future camera design. This would potentially power data-driven design.

No previous analysis of these temperature time series has been performed, and therefore the possible forecasting horizon and accuracy are unknown. As the YLC is dependent on temperature, the quality of temperature data is crucial for making lifetime predictions. Consequently, this work will also focus on methods for increasing the quality of the temperature data by interpolating missing values. Simple techniques such as linear interpolation as well as the possibility of developing and applying a more advanced interpolation method will be investigated and evaluated.

1.2 Purpose

The purpose of this master thesis is to support data-driven design decisions of camera capacitor components. This will be achieved by investigating the lifetime of cameras with respect to long term heat exposure in capacitors. Since a substantial amount of temperature data is missing, a sub-goal is to improve the data quality by different interpolation techniques. The main goal is to obtain a quantitative assessment of how well the yearly lifetime consumption (YLC) can be predicted and find a threshold for the amount of data required to achieve predictions with a certain accuracy. The predicted YLC will be used to describe the heat exposure of an extended camera population, providing knowledge for more refined design decisions.

2

Theory

2.1 Component Lifetime

The lifetime of an electrolytic capacitor depends on electrical parameters such as ripple current[1] and operating voltages. It also depends on environmental variables such as humidity, temperature and vibration [Cherry et al., 2018]. In this work the lifetime is assumed to only depend on thermal stress. An argument for making this assumption is that some of the previously mentioned effects might actually cancel to some extent. The effect of ripple current can actually extend lifetime in some ranges according to [Parler and Dubilier, 2004], while humidity and vibrations can decrease lifetime [Cherry et al., 2018]. Although, the effect of making this assumption has not been studied.

The lifetime equation used is derived from the Arrhenius equation [Chesworth, 2008], first described by Svante Arrhenius in 1889, explaining the temperature dependence of reaction rates in physical chemistry. From the Arrhenius equation, it is possible to derive an equation describing the lifetime of electrolytic capacitor components [Gupta et al., 2018], [Bocock, n.d.]

Definition 1. *Lifetime (L). Given an input temperature T, the lifetime L denotes the period in time (hours) that the capacitor component is expected to be functional, defined as*

$$L(T) := L_0 \, 2^{\frac{T_0 - T - T_{\text{offset}}}{c}}. \tag{2.1}$$

The component specific constants are

- $L_0 \in \mathbb{R}_{>0}$, lifetime at temperature $T = T_0$ - T_{offset},
- $c \in \mathbb{R}$, a constant depending on the capacitor,

[1] Ripple current is the voltage deviation that occurs when converting an alternating current to a direct current.

- $T_{\text{offset}} = T - T_{\text{sensor}}$ is the temperature (°C) offset between component and sensor. In this work it is assumed to be equal to three.
- $T_0 \in \mathbb{R}$ is the reference temperature (°C).

Eq. (2.1) for a component is only valid right after that component was manufactured. However, the rate of lifetime consumption is always the inverse of $L(T)$. The constants are specific to the capacitor used in each product, they are presented in Appendix A Table A.1.

Let T_t be the temperature at time t then definition 1 allows for calculation of a new metric called lifetime consumption.

Definition 2. *Lifetime Consumption (LC). Given a temperature vector* $\mathbf{T} = [T_j \dots T_i]$ *where temperature is sampled once per hour at times t where* $j \leq t \leq i$. *Lifetime consumption is the fraction of the component's total lifetime that has been consumed, defined as*

$$\text{LC}(\mathbf{T}) := \sum_{t=j}^{i} \frac{1}{L(T_t)}. \tag{2.2}$$

The entire camera life is consumed when LC reach 1. As will be clear later in this work, the temperature data is only available for a much shorter time period than the actual lifetime of the components. The lack of data describing the full life cycle of actual camera components would make predicting component lifetimes into an unsupervised regression problem. However, the lifetime prediction is reduced to a supervised regression problem by making an observation and an assumption; From the data, it seems very plausible that weather temperature has a significant impact on the camera temperature (observation). By assuming that the temperature profile is similar from year to year, it is possible to calculate the actual lifetime based on data from one year. Some cameras are located indoors, for those the temperature are more stable over time making the prediction more accurate. Based on this assumption the yearly lifetime consumption is formed.

Definition 3. *Yearly lifetime consumption (YLC). Given a temperature vector* $\mathbf{T}_N = [T_1 \dots T_N]$ *of length N the fraction of the lifetime that has been consumed over a year* [2] *is approximated as*

$$\text{YLC}(\mathbf{T}) := \frac{8760}{N} \sum_{t=1}^{N} \frac{1}{L(T_t)}, \tag{2.3}$$

[2] The effect of leap year will be disregarded.

where 8760 is the number of hours per year. E.g if four months of data is available, the sum is multiplied by a factor three. The factor $\frac{8760}{N}$ is necessary to compensate for lack of data. With this factor the YLC can be approximated regardless of how much data is available. It will be clear later that there is a large variety in the amount of temperature data available. In fact the available data is always less than one year, there are no time series that are one year long.

For visualisation and a more intuitive understanding of lifetime consumption a new metric is defined:

Definition 4. *Yearly Time to Live (*YTL*). The lifetime of a component given a* YLC *is defined by*

$$\text{YTL} := \text{YLC}^{-1}. \tag{2.4}$$

YLC is the fraction of the component's total lifetime spent per year, hence YTL is the number of years the component would last if YLC was constant from year to year.

Cameras with a short life are especially interesting from a product design perspective. For those the YLC value is large and the YTL value is small. Therefore YLC is a better measure of the degradation of a camera component and is used in calculations. When computing errors in estimating the YLC or YTL values, the mean square error and mean absolute error penalises larger values more. YTL is better for visualisation since it tells directly how long a component will be functioning.

2.2 Data Pipeline

Before the data from the cameras can be analysed it flows through a data pipeline, which involves both data lakes and databases. Fig. 2.1 below illustrates this entire process.

The data lake makes it possible to store data of any type. It can be unstructured (such as emails or PDFs), semi-structured (like CSV-files or JSON-files) or structured (rows and columns) like it is in a database.

The data collection process starts with a request for data from a camera individual once per hour per connected camera. However, to avoid a momentarily increase of bandwidth for locations with a large number of cameras, the time of the request is selected at random every hour for every camera. The 'raw' data, extracted from the cameras, is unstructured and is initially placed in a data lake hosted by a cloud storage provider. This data is then pre-processed and stored in another data lake (with semi-structure). It is then structured and moved to a third data lake. This storage solution is used for long term storage and contains all of the collected data, whereas the first two only contain data form the past two weeks.

From the long term storage solution, the data is transferred via an application called 'data transfer box' (DTB). The DTB type casts the data and moves it to another long term storage solution located at Axis as well as a search database (Elasticsearch). The search database only contains data from the past three months and is used for quick analysis or visualisation. The long term storage unit is called 'onprem'. It contains all the collected data and is where the data in this work is extracted from.

2.3 Data

Axis communications has approximately 200 different camera products and sell approximately three million cameras per year. Only a small subset of these cameras provide (non-personal) data for Axis.

Let the temperature T be a random variable

$$T : \Omega \to \mathbb{R},$$

drawn from some distribution and T^* be a realisation of that random variable. The distribution function is then given by $F_T(T^*) = \mathbb{P}(T \leq T^*)$. Differentiating gives the probability density function $f_T(T^*) = \frac{d}{dT^*} F_T(T^*)$.

For camera individual i and time t, a realisation of a random variable is a random variate $T^*_{i,t}$. For individual i, a vector describing the temperature process as a function of time is given by $\mathbf{T}_i = [T_1 \quad \dots \quad T_{N_i}]$, where individual i has N_i data points. When these vectors are stacked, they form a two-dimensional array $\mathbf{T}$, given by

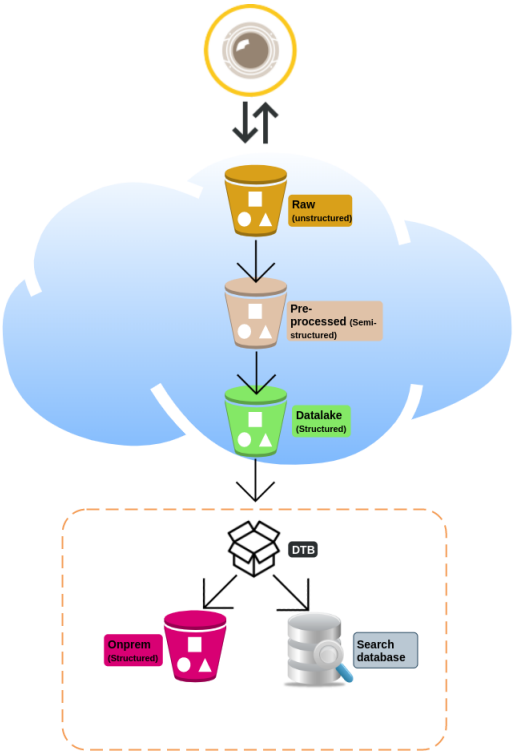


Figure 2.1 Axis's data pipeline. The cloud represents cloud storage and the orange rectangle represents storage at Axis. The yellow circle represents the cameras.

$$\mathbf{T} = \begin{bmatrix} T_{1,1} & T_{1,2} & \dots & T_{1,N_1} \\ T_{2,1} & T_{2,2} & \dots & T_{2,N_2} \\ \vdots & \vdots & \ddots & \vdots \\ T_{M,1} & T_{M,2} & \dots & T_{M,N_M} \end{bmatrix}, \tag{2.5}$$

where M is the number of individual cameras. From here on, in the interest of simplicity, all lengths are denoted N.

2.4 Data Quality

In order to get good accuracy in predicting the lifetimes it is necessary to have good quality of the temperature data which forms the basis of this analysis. However, there are several problems with the data:

1. Sampling frequency,
2. Jitter,
3. Measurement errors,
4. Missing values,
5. Interruptions in time series,
6. Short length of the time series.

To get good accuracy in calculations and predictions of the YLC, it is necessary to deal with these problems. Here it is outlined how these problems arose and how they were handled. The problems 1, 3 and 6 cannot be controlled but are explained for completeness.

1. The temperature is sampled, i.e. a reduction of a continuous signal to a discrete signal is made. If the sampling frequency is low, it creates distortions. Restrictions in storage and network bandwidth create restrictions in sampling frequency. The data is sampled once per hour (sampling frequency $2.78 \cdot 10^{-4}$ Hz). If data was available more often, the definition of lifetime (definition 1) would have to be reworked and could then be more accurate.

2. Axis wishes to limit the stress on a customer's internet connection. Hence the data is not collected all at once because this might cause a spike in bandwidth when a customer has many cameras at the same location. Instead it is sampled during the hour, using a random delay in retrieval time. This creates a jitter in the signal. Jitter is the sampling deviation from true periodicity of a periodic signal.

   To test the effect of the jitter, a simple test is performed. For all time series, a new data set is formed by adding an artificial jitter (AJ) to each temperature in $\mathbf{T}$. The jitter is added by replacing each temperature point T_t by

   $$T_t^{\mathrm{AJ}} = T_t + d(T_{t+1} - T_t)$$

   where $d \sim U(0,1)$ creating $\mathbf{T}^{AJ}$. For all time series YLC^{AJ} is calculated from the new data set and then compared with YLC_N from $\mathbf{T}$. The effect of jitter is studied using the mean absolute normalised error between true YLC values and YLC values with an added artificial jitter as

   $$\mathrm{MANE}^{\mathrm{AJ}} = \frac{1}{M} \sum_{i=1}^{M} \left| \frac{\mathrm{YLC}_{i,N} - \mathrm{YLC}_{i,N}^{\mathrm{AJ}}}{\mathrm{YLC}_{i,N}} \right|. \tag{2.6}$$

3. In all measurements there are errors, often modelled as an additive normally distributed noise. The measurement noise is considered an insignificant problem.

4. The time series contains significant amounts of missing data. This is considered the most serious data complication; every point needs to have a value. Fortunately, it is possible to amend this problem by interpolation. This issue will be discussed in detail later. When a missing value is substituted by an interpolated value an error is created.

   There are different types of missing values, the type of missing values here are considered to be missing completely at random (MCAR). Missing values are MCAR if the events that lead to the particular data point being missing are independent of the observable variables and unobservable parameters of interest and occur entirely at random [Seaman et al., 2013]. The main reason for missing data is considered to be internet connection failure. This is independent of temperature, and therefore analysis performed on data is unbiased. Since the data are time series, it is crucial to interpolate the missing data points. Each timestamp needs a value; otherwise, it is not possible to perform the analysis.

5. Interruptions in the time series occur, for instance, when a camera is turned off and then turned back on. This also creates missing values. Fig. A.1 in the Appendix A shows a histogram of the lengths of interruptions.

6. The length of the time series varies substantially among different individuals. For some individuals there are 11 months of temperature data is available. For all others there is less data available. Therefore there are never any true YLC-values, instead a compensating factor is introduced in the equation defining YLC, definition 3 and Eq. (3).

2.5 Data Preprocessing

Concatenation of time series

It is important to analyse the effect of the interruptions in the time series. Suppose the analysis concludes that the interruptions do not cause significant changes in the temperature process. In that case, it is possible to concatenate the time series, and interpolate the missing values, which would increase both the size and the value of the time series.

For camera i, $T_{i,t_1:t_r}$ and $T_{i,t_{r+h+1}:t_N}$ are two time series with an interruption in between them at time t_r, lasting h hours. The following differences are formed.

$$\Delta T_1 = T_{i,t_{r+h+1}} - T_{i,t_r}, \tag{2.7}$$
$$\Delta T_2 = T_{i,t_r} - T_{i,t_{r-h-1}}. \tag{2.8}$$

ΔT_1 denotes the temperature difference after and before the interruption. ΔT_2 denotes the difference between the temperature before the interruption and the temperature h steps before the interruption. ΔT_2 is created as a fair comparison to ΔT_1. ΔT_1 and ΔT_2 was computed for multiple interruption events, in order to obtain a distribution of the differences. If there isn't a significant difference between the distributions, it is confirmed that, generally, no change in the temperature process existed for interruptions in the time series.

Box-Cox transformations

A common technique that can be used to stabilise variance and make the data more normally distributed is the Box-Cox transformation [Box and Cox, 1964], which is defined as:

$$y_t^{(\lambda)} = \begin{cases} \lambda^{-1}\left(y_t^{\lambda} - 1\right) & \lambda \neq 0, \\ \log\left(y_t\right) & \lambda = 0, \end{cases}$$

where in general y_t may be any numeric data and λ is determined from maximising the log-likelihood:

$$L(\lambda) = -\frac{N}{2}\log\left\{\hat{\sigma}_y^2(\lambda)\right\} + (\lambda - 1)\sum_{t=1}^{N}\log\left(y_t\right),$$

where $\hat{\sigma}_y(\lambda)$ is the estimated standard deviation of the transformed data, using the parameter λ [Jakobsson, 2013]. This technique is applied to the target values when training and predicting YLC in Random forest models. Before any other processing such as error calculations and graph plotting, the data is transformed back using the inverse transformation given by

$$y_t = \begin{cases} \left(\lambda y_t^{(\lambda)} + 1\right)^{\frac{1}{\lambda}} & \lambda \neq 0, \\ e^{y_t^{(\lambda)}} & \lambda = 0. \end{cases}$$

2.6 Interpolation Techniques

In mathematical optimisation, a loss function is a function that maps a value onto a real number representing some loss connected with the value. In this setting this

value is the temperature interpolation value. Typically the interpolation technique is chosen based on which one minimises the loss functions mean square error (MSE) and/or mean absolute error (MAE) the most. The MSE is given by

$$\text{MSE} = \frac{1}{M}\frac{1}{|\mathbb{D}_i|}\sum_{i=1}^{M}\sum_{t\in\mathbb{D}_i}\left(T_{i,t} - \hat{T}_{i,t}\right)^2 \tag{2.9}$$

where $T_{i,t}$ is the true temperature and $\hat{T}_{i,t}$ is the imputed value of $T_{i,t}$ when it is missing, and $\mathbb{D}_i = \{t \in \mathbb{R}, T_{i,t} \text{ is missing}\}$ is the set of the indices of missing values for time series i.

The mean absolute error is easier to interpret. It yields the mean temperature difference per missing point, and is defined as:

$$\text{MAE} = \frac{1}{M}\frac{1}{|\mathbb{D}_i|}\sum_{i=1}^{M}\sum_{t\in\mathbb{D}_i}\left|T_{i,t} - \hat{T}_{i,t}\right| \tag{2.10}$$

To evaluate the performance of different interpolation techniques, two data sets will be used. One with time series without sequences of missing values, and one with the same time series but with sequences of removed data points with random starting points. The length of the sequences will be the same for every time series. The second data set will then be interpolated. This procedure will then be conducted with many different lengths of the sequences. The selection of interpolation technique will be based on the MSE and the MAE, and the chosen one will be used to impute missing values throughout the work.

Linear interpolation

Linear interpolation is a simple method, for each value that is missing, T_t is given by the following formula (2.11), taking the closest preceding point (t_a, T_a) and the closest succeeding point (t_b, T_b) of the missing value as input.

Definition 5. *Linear interpolation. Given a preceding point (t_a, T_a) and succeeding point (t_b, T_b) the linear interpolation is given by*

$$\hat{T}_t = T_a + (T_b - T_a)\frac{t - t_a}{t_b - t_a}. \tag{2.11}$$

Naive interpolation

There is 24 hour periodicity in the temperature time series, therefore another technique that could be used is a 24-hour interpolation, named 'naive interpolation'. The naive interpolation replaces the missing value with the value that was 24 hours earlier. If missing value appears within the first 24 hours linear interpolation is used. It is defined as

Definition 6. *Naive interpolation. The value at time t is given by the value at time t - 24 if $t \geq 24$, otherwise linear interpolation is used were (t_a, T_a) is the preceding point and the succeeding point (t_b, T_b).*

$$\hat{T}_t = \begin{cases} T_{t-24}, & \text{if } t \geq 24 \\ T_a + (T_b - T_a)\frac{t-t_a}{t_b-t_a}, & \text{otherwise} \end{cases}$$

This interpolation technique will always have a value to insert if there are values in the first 24 positions, which are guaranteed by using linear interpolation.

The Vallier interpolation method

To obtain a more sophisticated method for imputing missing values, a combination of the naive predictor and linear interpolation called *Vallier interpolation* was developed. This method aimed to capture a sudden increase in day temperature. The method utilises the two temperatures before and after the segment of missing values, the temperatures 24 hours prior to them, and the temperatures 24 hours prior to the missing values.

Consider a time series with a segment of h consecutive missing values. Let the index of the value before the segment be denoted t_0 and the index of the value after the segment be denoted t_1, i.e.

$$T_{t_1} = T_{t_0+h+1}.$$

Let the estimate of the temperature at time $t_0 + r$ be denoted $\hat{T}_{t_0+r}$ for $1 \leq r \leq h$. Then let the temperature T_{t_0-24+r} be the temperature 24 hours prior to T_{t_0+r}, and consider the two temperature differences:

$$\Delta^{\text{FOR}} = T_{t_0-24+r} - T_{t_0-24}, \tag{2.12}$$
$$\Delta^{\text{BACK}} = T_{t_1-24} - T_{t_0-24+r}, \tag{2.13}$$

where T_{t_0-24} is the temperature 24 hours prior to T_{t_0}, and T_{t_1-24} is the temperature 24 hours prior to T_{t_1}.

From Δ^{FOR} and Δ^{BACK} the temperatures $T^{\text{FOR}}_{t_0+r}$ and $T^{\text{BACK}}_{t_0+r}$ are defined as:

$$T^{\text{FOR}}_{t_0+r} = T_{t_0} + \Delta^{\text{FOR}},$$
$$T^{\text{BACK}}_{t_0+r} = T_{t_1} - \Delta^{\text{BACK}},$$

where $T^{\text{FOR}}_{t_0+r}$, and $T^{\text{BACK}}_{t_0+r}$ is represented in Fig. 2.2 by the green and blue curve, respectively. Note that the estimate of T_{t_0+r} by $T^{\text{FOR}}_{t_0+r}$ becomes less accurate as r increases and vice versa for $T^{\text{BACK}}_{t_0+r}$.

The final estimate at t_0 is obtained by a combination of the two forecasts, $T^{\text{FOR}}_{t_0+r}$, and $T^{\text{BACK}}_{t_0+r}$. The closer the time is to t_0 (i.e. for small r when $t = t_0 + r$), more weight is given to the estimate $T^{\text{FOR}}_{t_0+r}$. Conversely as time approaches t_1, more weight is given to $T^{\text{BACK}}_{t_0+r}$. The estimate is defined as:

$$\hat{T}_{t_0+r} = \frac{(h-r+1)T^{\text{FOR}}_{t_0+r} + rT^{\text{BACK}}_{t_0+r}}{h+1}, \tag{2.14}$$

and the curve of the estimated temperatures is represented by the red curve in Fig. 2.2.

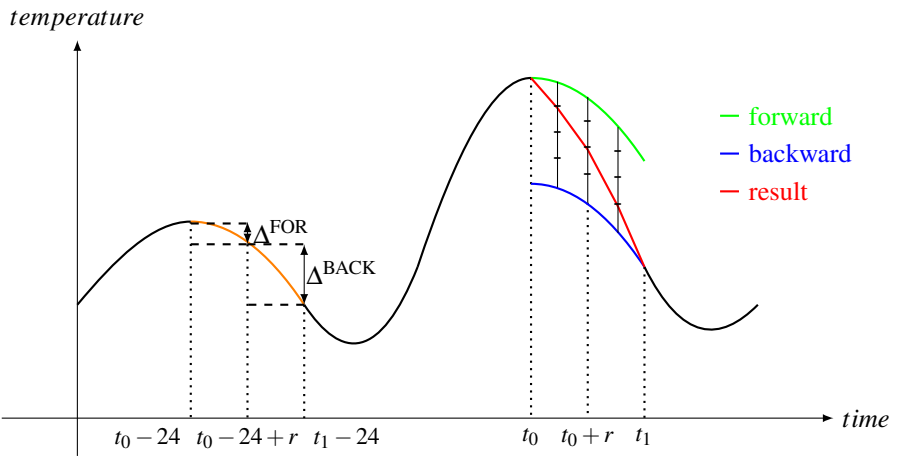


Figure 2.2 An illustration of the Vallier interpolation.

If $h \geq 24$, it is not possible to compute Δ^{BACK} for any of the values as the value T_{t_1-24} in equation (2.13) is missing. Therefore, T_{t_1-24} is replaced by the forward estimate $T_{t_0} + \Delta^{\text{FOR}}$, which is then used for computing Δ^{BACK}. Then the two values for Δ^{FOR} and Δ^{BACK} are used according to Eq. (2.14) to interpolate the missing values.

2.7 Validation of the Target

Since the available data is limited and does not contain a full year of temperature data for any individual, the true YLC value is unknown. Therefore, YLC calculated with a factor compensating for the lack of data will be used as the target value for the machine learning model. An analysis is performed to control and quantify the error in the target variable that is created from lack of data. The aim is to keep all the time series that can deliver a target value within $\pm 5\%$ of the true value and discard the rest. The YLC will be estimated for all cameras with 11 months of time series data. As defined by Eq. (2.3), which multiplies the 11 months YLC by a factor $\frac{12}{11}$. The YLC will then be compared to other estimated YLC values computed from different amounts of data varying from 1 to 10 months.

It is of interest to know how the YLC varies, depending on which month/months that is chosen for the computation. Therefore, the YLC will be estimated for every possible combination of consecutive months for every data amount (1-10 months), e.g., the YLC computed from one month of data will be computed 11 times, once for every possible month. Fig. 3.7 in section 3.2 shows these results for one arbitrary time series from the camera model M3045-V. To calculate the error of the target value, the mean absolute normalised error (MANE) was used. It is for an individual,

$$\mathrm{MANE^d} = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{\mathrm{YLC}_N - \widehat{\mathrm{YLC}}_{d,k}}{\mathrm{YLC}_N}\right|, \tag{2.15}$$

where

- d denotes the amount of data in months,
- YLC_N is the YLC with 11 months of data,
- $\widehat{\mathrm{YLC}}_{d,k}$ is the YLC computed with d amounts of data for one of the K possible combinations.
- $K = 12 - d$ denotes the number of different YLC calculations that can be done.

Then the average of MANE over all individuals for each camera product is calculated as:

$$\mathrm{MANE}_p^d = \frac{1}{\mathrm{I_p}}\sum_{i=1}^{\mathrm{I_p}}\mathrm{MANE}_i^d, \tag{2.16}$$

where I_p denotes the total number of individuals for a camera product p.

Since the underlying capacitor constants and distribution of the temperature data vary with product, the time series length requirement to accept a target value of YLC will be specific for each camera product. The required length of the time series for camera product p will be determined based on MANE_p^d. The time series length requirement is the shortest length d that has a $\text{MANE}_p^d < 0.05$.

2.8 Random Forest

The Random forest algorithm [Breiman, 2001] is an ensemble machine learning method used for classification and regression. It utilises an aggregation of the results from many decision trees to create a prediction.

Each tree is constructed from a bootstrapped data set, which is of the same size as the original data set. It is created by drawing samples with replacements from the original data set [Friedman et al., 2001].

A tree is constructed by splitting the data set multiple times based on different features until some stopping criterion is reached. For instance, until each terminal node contains at most a pre-selected minimum number of samples. Fig. 2.3 illustrates a simple decision tree for classification.

Before each split, a selected number of random features are considered. The feature which yields the best tree, depending on some criterion, is used for the separation. For classification, it is usually Gini impurity, and for regression, it is often the mean square error [Cutler et al., 2012].

This process is then repeated B-times, which yields B different decision trees. For regression the final output $f_{\text{rf}}^B(x)$ is obtained by averaging over all the trees [Friedman et al., 2001], i.e.

$$f_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} E_b(x),$$

where B is the number of trees and $E_b(x)$ is the output of a particular tree.

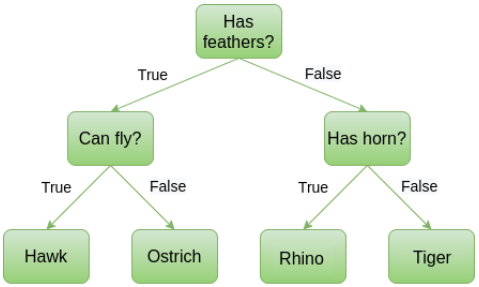


Figure 2.3 A simple decision tree example. The leaf nodes contains the animals.

Hyperparameters

Compared to many other ML-algorithms, Random forest does not necessarily require a lot of tuning [Probst et al., 2018]. However, some tuning is usually necessary [Fernández-Delgado et al., 2014]. Although optimisation of all the hyperparameters would be preferable, only some will be tuned in this work, as many of them are highly correlated. The first one is the number of trees (B) used for constructing the model. The second one is the number of random features the model should consider before each split, and the third one is the minimum node size. Furthermore, the size of the bootstrapped data set will be tuned as well as whether or not to bootstrap the data set. The other changeable hyperparameters are the max depth of the tree, the number of samples required to perform a split, and the separation criterion. For these three the default values will always be used. The parameters which will be tuned will be denoted:

- B - The number of trees in the forest.
- D - The minimum number of samples in the final nodes (leaf-nodes).
- F - The number of random features considered before each split.
- S - Sample size of the bootstrapped data set.

The number of trees in the forest Increasing the number of trees generally won't increase the risk of overfitting [Breiman, 2001] and the number of trees should be set as high as computationally feasible [Probst and Boulesteix, 2017]. Therefore, the number of trees will be selected by balancing the run time and the results.

The number of random features The number of random features to consider before each split has been shown to generally be the most important feature to optimise with regard to performance [Hutter et al., 2014], [Probst et al., 2019]. Studies have shown that an increase in performance is obtainable by only optimising this parameter [Bernard et al., 2009], [Dıaz-Uriarte and De Andres, 2006]. However, the

possible gain in performance varies between different data sets. If the number of relevant features is low compared to the total number of features, the model could be trained on irrelevant features making it perform worse. On the contrary, using a smaller F decreases the correlation between the trees, making the model more stable when averaging over the trees. Generally for regression, the default value of F is [Friedman et al., 2001]:

$$F = \frac{\text{total number of features}}{3}.$$

Minimum number of samples in the final node The depth of the tree is directly correlated to the selected number of minimum samples in the final nodes. A low value yields a deeper tree and vice versa. For a single tree, the bias will decrease with a decreasing sample size in the final nodes, but the variance will increase. Averaging over multiple trees will reduce this variance but usually for regression, the minimum node size in the leaf nodes is set to five [Friedman et al., 2001]. However, [Segal, 2004] showed that tuning this parameter can boost the result of the model.

Bootstrapping It is possible to use the original data set when building the trees for the model and skip the procedure of creating a bootstrapped data set for every tree. Then, only the selection of a random subset of features before each split contributes to the randomness of the model, as every tree is built from the same data set.

Sub sample size If bootstrapping is used, one could bootstrap a smaller data size than the original one to create even more diverse trees.

Hyperparameter optimisation

A basic way of finding the optimal hyperparameters is by performing a grid search. A grid is constructed from manually selected ranges for the hyperparameters. A parameter sweep is then conducted, and a model is built with every possible combination of the hyperparameters. The one which obtained the best results is chosen. Usually, cross validation is used for evaluating the results. As the dimensions of the grid increase, this approach becomes computationally expensive. An alternative is a randomised search over the grid. The hyperparameters are then drawn at random, and the number of tries is selected based on computation capacity. Of course, many other, more sophisticated methods exist, such as sequential model-based optimisation [Jones et al., 1998]. Sequential model-based optimisation has been proven successful for hyperparameter optimisation for Random forest [Probst et al., 2019]. Since there is sufficient computational power to perform a grid search, this technique will not be further investigated.

Data Partitioning, Overfitting and Cross Validation

A model can capture the structure of the data too well. The variance of the model then becomes very high, which might result in overfitting. The most established

way to detect and avoid overfitting is to split the data into different subsets. These subsets are usually called training set, validation set, and test set. The training set is used for training the model and contains around 70-80 % of the available data. The validation set is used for selecting the hyperparameters and the features and contains 10-20 % of the data. The test set is only used for evaluating the final model and is never used during the development of the model. A significantly lower accuracy or higher error on the test set or the validation set compared to the training set is an indication of overfitting.

An alternative to the validation set is to use cross validation. The training data is then divided into K folds. $K-1$ of the folds are then used for training the model, and one is used for validation. The procedure is then repeated K-times, where a different fold is used for validation each time. The results from the validation folds can then be used for determining the hyperparameters in the model.

Concept drift

The fundamental assumption in machine learning is that the training data is independent and identically distributed (i.i.d). Concept drift is an example of an i.i.d. violation when data changes over time.

Suppose the statistical properties of the target values or the relationship between the features and the target changes over time, in an unforeseen way. In that case, the data is said to contain a concept drift. To partly mitigate concept drift and to be able to quantify the effect of concept drift on the model accuracy, a small amount of not i.i.d. samples taken from the training set could be placed in a sub-set. The sub-set could be used for the all selections during model development and as an additional test set to quantify the accuracy of the final model.

Feature selection

When constructing a machine learning model, it is always a challenge to only include relevant features. Reducing the number of features in the model will reduce the model's training time, the complexity of the model, and the risk of overfitting. If the right set of features is used, the accuracy can increase [Aha and Bankert, 1996]. Feature selection methods are usually divided into three main categories; filter methods, embedded methods, and wrapper methods [Kohavi, John, et al., 1997], [Guyon and Elisseeff, 2003]. Wrapper methods utilise the machine learning algorithm which one wishes to use. These methods involve training a new model for many different subsets of features and then compare the trained models based on some evaluation criterion, e.g., R^2-value [Kohavi, John, et al., 1997]. The subset, which yields the best model based on the evaluation criterion is chosen. How the possible sub-sets of features are chosen varies from different wrapping techniques. As wrapping methods require training of multiple models, they are the most computational expensive feature selection approach. The training of numerous models also makes wrapping methods more prone to overfitting.

Filter methods are independent of any machine learning algorithm. The features are selected before the training of the model and are chosen based on some statistical correlation score between the feature and the target value. Consequently, the features are selected without regard to how the selected subset of features impacts the outcome of the machine learning model [Kohavi, John, et al., 1997].

The last of the three embedded methods, perform feature selection during the training of the model. They are therefore limited to machine learning algorithms, for which it is possible to do a feature importance evaluation during the training. Random forest is such an algorithm. It is possible to evaluate the importance of a feature when the trees are constructed. As embedded methods only require training of one model, they require significantly less computing power [Guyon and Elisseeff, 2003].

Forward sequential selection Forward sequential selection belongs to the category of wrapper methods for feature selection. First, a model is constructed from only one feature for every single feature. The models are then evaluated based on some metric. Secondly, the feature which gave the best model is used together with one of the remaining features. A model is developed with every remaining feature, and the model which obtained the best results with two features is selected. The same procedure is then repeated, until the adding a feature does not give a model which performed better than the previously best model. The features which was included in the best overall model are used for the development of the final model [Aha and Bankert, 1996].

Possible features

Many of the features will be computed from a shortened version of the original time series. This is because it is of interest to study how features extracted from shorter time series may predict the YLC. The shortened time series will have the same starting point (t = 1) as the original one. This series is a vector of n temperature data points given by $\mathbf{T}_n = [T_1 \dots T_n]$ where n is the number of data points used in the shorted time series. Therefore $\mathbf{T}_n$ contains the same data as $\mathbf{T}$ defined in Eq. (2.5) truncated to n data points. From this vector of temperature data, many features are calculated to investigate how specific transformations of temperature might predict YLC.

Recall the YLC definition 3, $\mathrm{YLC}_n(\mathbf{T}) := \frac{8760}{n} \sum_{t=t_1}^{t_n} \frac{1}{L(T_t)}$ this equation is used for what is called "YLC-transformation". This transformation takes an arbitrary number of temperature data points and returning an estimated YLC-value. This is written as $\mathrm{YLC}_n(\cdot)$, where $(\cdot)$ may be for example mean, where the mean value would be the only input value (n = 1) giving

$$\mathrm{YLC}_n \text{ mean} = 8760 \frac{1}{L(\bar{T}_n)},$$

since the mean temperature is given by $\bar{T}_n$. n gives the length of the temperature vector. All features and their description are shown in Table 2.1.

Feature selection approach The feature selection will be conducted with one particular length of the short time series, and with the features shown in table 2.1, as a re-evaluation of the features for every n is considered too complicated. In addition, a feature containing random numbers will be included in the feature selection process, mainly as a confirmation that the other features contributes to the model.

Table 2.1 Description of the features which will be used in the feature selection. The features are shown for one individual. Here $\mathrm{YLC}_n(\cdot)$ denotes a YLC transformation of $(\cdot)$. Target* - YLC_N is the target variable. Baseline model** - YLC_n is the YLC-transformation of all the temperature feature data, it is denoted as "Baseline model" throughout this work.

Feature	Description	Type
Target*: YLC_N	The YLC computed on $\mathbf{T}_N$.	Float
T Standard deviation	The standard deviation of $\mathbf{T}_n$.	Float
T mean	The mean of $\mathbf{T}_n$.	Float
T max	The max of $\mathbf{T}_n$.	Float
T min	The min of $\mathbf{T}_n$.	Float
Baseline model**: YLC_n	The YLC computed from $\mathbf{T}_n$.	Float
YLC_n mean	The YLC computed from T mean.	Float
YLC_n max	The YLC computed from T max.	Float
YLC_n min.	The YLC computed from T min.	Float
Month	The month of the first value in $\mathbf{T}_n$.	Integer
c	c constant in Eq. (1).	Integer
L_0	L_0 constant in Eq. (1).	Integer
T_0	T_0 constant in Eq. (1).	Integer
T q10	The 10th quantile of $\mathbf{T}_n$.	Float
T q90	The 90th quantile of $\mathbf{T}_n$.	Float
YLC_n q10	The YLC computed from T q10.	Float
YLC_n q90	The YLC computed from T q90.	Float
Product	The product type.	Integer
IR (Infrared radiation)	If the camera has capability of creating images from infrared radiation.	Boolean
Audio detection	If the camera can detect audio.	Boolean
Outside	If the camera is built for outdoor environment.	Boolean
ISS (Image sensor size)	The size of the image sensor in the camera. (Inches.)	Float

Model selection

The R^2-value or R^2-score of predictions based on model are defined as

$$R^2 = 1 - \frac{\frac{1}{M}\sum_{i=1}^{M}(\mathrm{YLC}_{i,N} - \mathrm{YLC}_{i,\mathrm{predicted}})^2}{\frac{1}{M}\sum_{i=1}^{M}(\mathrm{YLC}_{i,N} - \overline{\mathrm{YLC}}_{i,N})^2}, \tag{2.17}$$

where

$$\overline{\mathrm{YLC}}_{i,N} = \frac{1}{M}\sum_{i=1}^{M}\mathrm{YLC}_{i,N}.$$

For selecting models the mean absolute normalised error is also used

$$\text{MANE}_\text{n} = \frac{1}{M}\sum_{i=1}^{M}\left|\frac{\text{YLC}_{i,N} - \text{YLC}_{i,\text{predicted}}}{\text{YLC}_{N,i}}\right|, \tag{2.18}$$

After the feature selection, the grid search will evaluate the different models based on the R^2-score from cross-validation on the training set and for the different values of n (length of the temperature vector). Following, one particular value of n will be selected and a subset of models with this n will be further analysed. They will be selected based on their R^2-score from the cross validation and their hyperparameters. Models with different hyperparameters will be included in the subset. A model with only standard hyperparameters will also be included in this subset. The accuracy on the validation set and test set will be tested for every model in this subset. They will also be compared to a Baseline model, which serves as a benchmark to beat when evaluating the performance of the Random forest models.

For the other values of n, only the models with the highest mean R^2-score from the cross validation will be presented.

2.9 Hypothesis Testing

It is of interest to know how the distribution of the YLC varies depending on which data set is used for the estimation. **ylc** denotes a vector of YLC-values. Three different data sets, $\mathbf{ylc}_n$, $\mathbf{ylc}_N$ and $\mathbf{ylc}_{RF}$ are available for this investigation. Here $\mathbf{ylc}_n$ and $\mathbf{ylc}_N$ are vectors of YLC-values calculated based on $\mathbf{T}_n = [T_1 \dots T_n]$ and $\mathbf{T}_N = [T_1 \dots T_N]$ respectively, according to Eq. (2.3). $\mathbf{ylc}_{RF}$ contains the predictions of the Random forest model.

Baseline model

In this work $\mathbf{ylc}_n$ is denoted as the Baseline model (BM). This provides a good benchmark to compare the other models to. If it is not possible to beat the benchmark, modelling with Random forest is not necessary.

Hypothesis testing - theory

Empirical distributions based on these vectors are formed. It is of interest to investigate to which degree these distributions are similar. In order to draw conclusions from the comparison some theory is necessary.

Let $F_X(x) = \mathbb{P}(X \leq x)$ be some cumulative distribution function (CDF) and $\hat{F}_{X,M}(x)$ be the empirical distribution function (EDF) constructed from M samples, sampled from the $F_X(x)$ distribution. Let $G_X(x)$ and $\hat{G}_{X,M}(x)$ be defined similarly for some other distribution. In general an EDF $\hat{F}_{X,M}(x)$ is defined as

$$\hat{F}_{X,M}(x) = \frac{1}{M}\sum_{i=1}^{M} 1_{\{X_i \leq x\}},$$

where

$$1_{\{X_i \leq x\}} = \begin{cases} 1 & \text{if } X_i \leq x \\ 0 & \text{otherwise} \end{cases},$$

is the indicator function of the event $X_i \leq x$.

Several pairs of EDFs will be compared and presented in the result section of this thesis. It is of interest to test whether two distributions have statistically close to having the same mean and more generally, to test the hypothesis that two empirical distributions are statistically close. These results give important insight into how much data is necessary to create a good estimate of YLC and its distribution.

Hypotheses

Two hypotheses are formed. Firstly the null hypothesis that the means of the two populations are equal and secondly that the two samples come from the same distribution. These hypotheses are tested with a paired t-test and a Kolmogorov-Smirnov test, respectively. More formally

$$H_0 : \mathbb{E}[F(x)] = \mathbb{E}[G(x)],$$
$$H_a : \mathbb{E}[F(x)] \neq \mathbb{E}[G(x)],$$

is the hypothesis that the two samples have the same mean. Here H_0 and H_a denote the null and the alternative hypothesis respectively. The hypothesis of same distribution is given by

$$H_0 : F(x) = G(x),$$
$$H_a : F(x) \neq G(x).$$

The p-value is the probability of obtaining results at least as extreme as the results actually observed, assuming that the null hypothesis is correct.

$$p = \mathbb{P}(X \geq x \,|\, H_0).$$

If the p-value is below the significance level α, the null hypothesis (H_0) is rejected, otherwise it is said that the test fails to reject the null hypothesis and the null hypothesis is kept in lack of other evidence. In this work $\alpha = 0.05$ is used.

T-test for paired samples

A t-test is any statistical hypothesis test in which the test statistic follows a Student's t-distribution under the null hypothesis. The paired t-tests assume that the mean of the two distributions are normal distributed. This assumption holds for large samples for non-normal distributions by the central limit theorem. It tests the hypothesis that two samples have the same mean. The samples are paired because both data set contain data from the same TS. The paired t-test is preferred since it is more powerful than the unpaired t-test. In short, the statistical power is the probability that the test correctly rejects the null hypothesis [Brownlee, 2018]. Evidently a more powerful test is preferred.

Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test (KS-test) can test whether two arbitrary distributions are the same [unknown, 2020]. It is a rank-order test that tests whether or not two empirical distribution functions $(F_n(x), G_n(x))$ can be considered to come from the same continuous distribution function $F(x) = G(x)$ [Hodges, 1958]. The test statistic is given by

$$D_n = \sup_{x \in \mathbb{R}} |F_n(x) - G_n(x)|,$$

which is the largest vertical distance between the EDFs.

3

Results

3.1 Camera Products

The camera products that were studied are listed in Table A.1 in Appendix A. That table also shows the accompanying constants relevant for calculating the YLC. The products were chosen based on the list of the products with most individuals with temperature data. The Fig. A.3 and A.4 in Appendix A show pictures of the products that were studied.

3.2 Data Quality - Results

Artificial jitter

Recall Eq. (2.6), $\text{MANE}^{\text{AJ}} = \frac{1}{M}\sum_{i=1}^{M}\left|\frac{\text{YLC}_{i,N}-\text{YLC}_{i,N}^{\text{AJ}}}{\text{YLC}_{i,N}}\right|$ comparing the YLC calculated from data with and without added artificial jitter. The mean absolute normalised error of 10^{-3} are noted. E.g. for a camera with a lifetime of 20 years an 0.1% error of YLC would be give an error in YTL of 7 days, which is considered negligible. This is comparable in size to disregarding the effect of leap year.

Interpolation results

For this analysis, only time series from the camera model M2026-LE-MK II were included. A total of 3986 different time series was considered when evaluating the interpolation methods. Fig. 3.1-3.2 show the performance of the different methods. Fig. 3.1 shows the MSE computed according to Eq. (2.9), and Fig. 3.2 shows the MAE computed according to Eq. (2.10). The Vallier interpolation method is best for most of the number of removed points. An arbitrary time series with interpolated data from each technique is shown in Fig. 3.3, and it is clear that the Vallier interpolation technique adequately follows a change in the mean temperature. Therefore, Vallier interpolation was used for interpolating all the missing values.

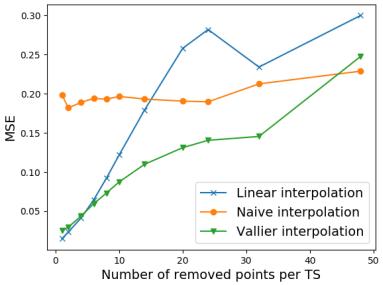


Figure 3.1 The resulting MSE for the different methods.

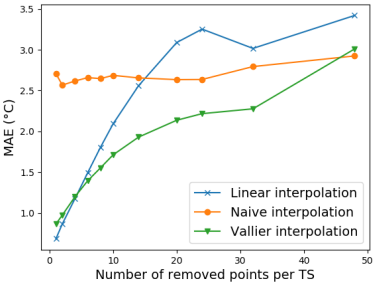


Figure 3.2 The resulting MAE for the different methods.

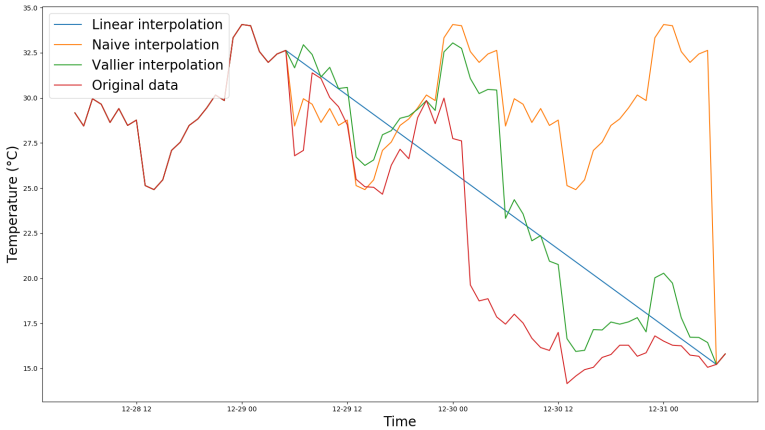


Figure 3.3 A visualisation of the different methods on an arbitrary time series with 48 missing values. The blue, orange, and green graphs are linear interpolated, naive interpolated, and Vallier interpolated data respectively. The red graph represents the original data.

Concatenated time series

The difference between the last temperature in a time series and the first temperature in the following one is called 'Temp diff between' and is given by Eq. (2.7). The temperature difference between the last value in the first time series and the temperature m steps back is called 'Temp diff within' and is given by equation (2.8). One thousand randomly selected gaps between time series from the product M2026-LE-MK II were included in this analysis. Fig. 3.4 shows a scatter plot of the

temperature differences and the gap lengths. Fig. 3.5 is a histogram of the temperature differences. Fig. A.1 in Appendix A shows a histogram over the length of the gaps.

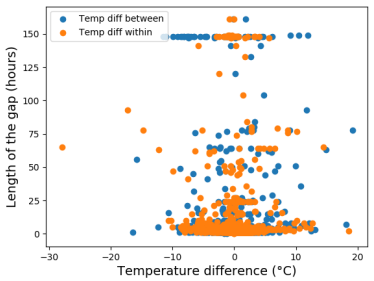


Figure 3.4 Scatter plot over the temperature differences defined by Eq. (2.7) - (2.8), and the length of the gaps.

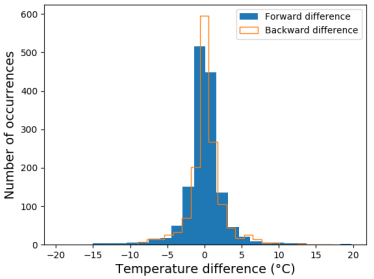


Figure 3.5 Histogram over the temperature differences defined by Eq. (2.7) - (2.8).

The two distributions of ΔT were compared with two statistical tests. If the distributions could be concluded to be the same, this indicates that generally, an interruption does not cause a change in the temperature process. Consequently, there is no distinction between the missing values caused by the interruptions and the regular missing values (point 4. in Section 2.4).

The paired t-test gave a p-value of 0.534 under the null hypothesis that the two distributions had the same expected value. Therefore, the null hypothesis was not rejected. The Kolmogorov-Smirnov test gave a statistic of 0.062, which led to a p-value of 0.043. Thus, the null hypothesis that two independent samples were drawn from the same continuous distribution was rejected.

Box-Cox transformation

From the training data the λ value was calculated to $\lambda = -0.1897$. Fig. 3.6 shows Box-Cox transformed target data. The data looks more normally distributed when it is transformed. This might improve the performance of the Random forest model but the effect was not studied.

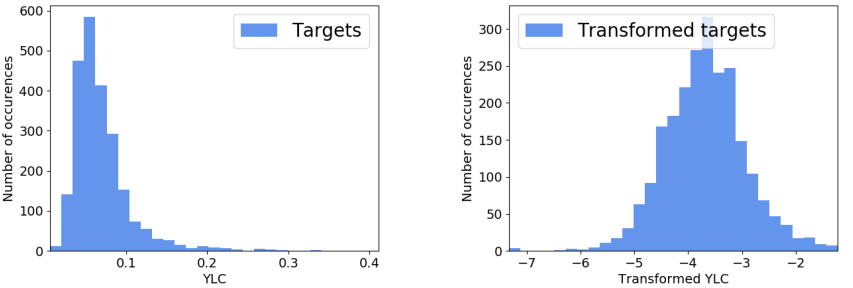
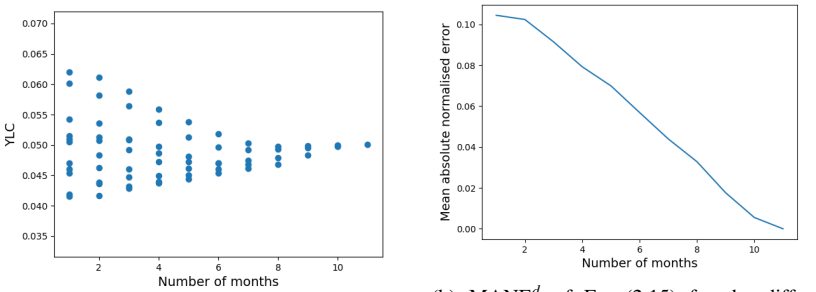


Figure 3.6 Comparison of the target YLC-values (left) and the Box-Cox transformed YLC-values (right). The right figure looks more normally distributed.

Target validation

The validation of the YLC for a single camera individual can be seen in Fig. 3.7. Fig. 3.7(a) shows the YTL computed from all the possible combinations of months for various amounts of data. Fig. 3.7(b) shows how the $\text{MANE}_{\text{M3045-V}}$ varies for the different number of months (Eq. (2.15)). Averaging the results in this plot over all the individuals from the same product resulted in the graph in Fig. 3.8, as described by Eq. (2.16). For the camera model M3045-V, this gave a $\text{MANE}_{\text{M3045-V}}$ of 5 % at 5.85 months. Thus, 5.85 months of data were determined to be sufficient for a target value for cameras of model M3045-V. Table 3.1 lists the resulting data requirements for all the other camera models as well as the number of available time series with 11 months of data.



(a) $\widehat{\text{YLC}}_{d,k}$ from different amounts of months.

(b) MANE^d of Eq. (2.15) for the different amount of months.

Figure 3.7 An arbitrary individual example of how the YTL can vary depending on which and how many months are used in the estimation. The time series used in this example was from the camera model M3045-V.

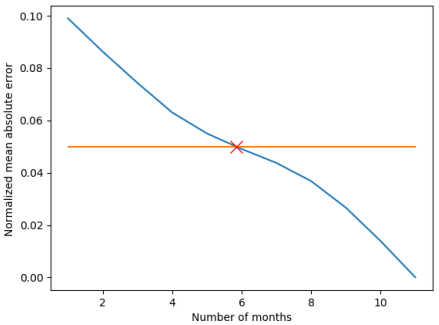


Figure 3.8 This figure shows the mean absolute normalised error from the product M3045-V Eq. (2.16). A MANE^d of 5 % was reached at 5.85 months, which means that 5.85 months of temperature data are needed for a time series from M3045-V to be used as a valid target value. Here 238 time series were used.

Table 3.1 The required length of the time series for different camera models in order to obtain a sufficiently accurate target value.

Product	**Required #month**	**Outdoor**	**#cameras**
M3045-V	5.85	No	238
M2026-LE-MkII	9.97	Yes	57
M3046-V	3.76	No	109
M3044-V	7.82	No	115
M3106-LVE-MkII	10.2	Yes	23
C 360 P	4.37	No	1
M3047-P	5.98	No	49
M2026-LE	9.02	Yes	32
M3048-P	2.88	No	5
C Dome WV	1.00	No	1
M3046-1-8mm	7.27	No	5

3.3 Random Forest

Data partitioning

A total of 2328 samples was used for the development of the models. 64 % of the samples were included in the training set, 16 % were included in the validation set and 20 % were included in the test set. The number of samples in each set can be seen in Table A.2 in Appendix A. The data partitioning was done completely at random. A non-IID data set, to study potential concept drift in the data was not included.

Feature selection

The sequential forward feature selection was conducted on the training set with the hyperparameters in Table 3.2, and the features shown in Table 2.1. The number of data points given to the model (n) can also be seen in the Table 3.2. The results of the feature selection can be seen in Fig. 3.9, and Table 3.3. The first 10 features were chosen to be further included in the development of the models. The models were evaluated based on the mean R^2-value of the 5 validation sets from 5 folds cross-validation on the training data.

Table 3.2 The hyperparameters used for the feature selection process. F varied between 1-7 as it was set to the total number of used features/3.

Hyperparameters	**Values**
Bootstrap	True
F	1-7
S	1
D	5
B	300
n	384 (16 days)

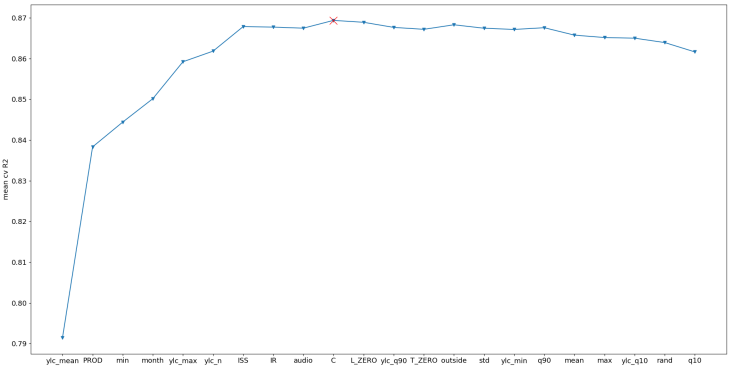


Figure 3.9 The mean R^2-score of the cross validation folds for the sequential forward selection method. The highest score is marked with a red cross and was obtain with the first 10 features. The ranking of the features can also be seen in Table 3.3.

Table 3.3 The resulting ranking of the features from the feature selection, see Fig. 3.9 for illustration.

Feature	Rank
YLC_n mean	1
Product	2
T min	3
Month	4
YLC_n max	5
YLC_n	6
ISS (Image sensor size)	7
IR (Infrared radiation)	8
Audio detection	9
C	10
L_0	11
YLC_n q90	12
T_0	13
Outside	14
T Standard deviation	15
YLC_n min	16
T q90	17
T mean	18
T max	19
YLC_n q10	20
Random	21
T q10	22

Hyperparameter search

The grid search was performed on the hyperparameter space shown in Table 3.4. Apart from the hyperparameters in Random forest, the table also shows the different amounts of data (n) the models were given. A total of 7840 models were developed for every n, and altogether 156800 models were built. The models were initially

evaluated based on the mean R^2-score from 5 fold cross-validation on the training set.

Table 3.4 The range of the hyperparameters for which the grid search was performed. For all possible combinations a Random forest model was constructed and its performance was evaluated using CV.

Hyperparameter	Grid
Number of estimators (B)	100, 400, 700, 1000, 2000, 3000, 5000
Number of features (F)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Min number of samples at leaf node (D)	3, 4, 5, 6, 7, 9, 12, 18
Bootstrap	True, False
Sub-sample size (S)	0.3, 0.5, 0.75, 0.85, 0.9, 0.95, 1
Number of data points given to the model (n)	1, 3, 6, 12, 24, 48, 72, 96, 144, 192, 288, 384, 480, 720, 960, 1440, 2160, 2880, 3600, 4800

Selected models

Four of the highest-ranked models with $n = 384$ (16 days) were chosen and further evaluated on the validation and test data set to check for overfitting. The models that were chosen are

- M1 - The highest ranked model
- M2 - The highest ranked model of the models which used bootstrapping.
- M3 - The highest ranked model of the models which used $B = 5000$.
- M4 - The highest ranked model of the models which used bootstrapping and $B = 5000$.
- M5 - A model that should perform well according to theory of Random forest.

The attributes of these models are shown in Table 3.5 and their performance on the validation and test set can be seen in Table 3.6. It is surprising that the models performs worse on the test set compared to the validation set since neither of the sets were used in the hyperparameters selection and samples were split at random.

Selected models for varying values of n

The models which obtained the highest mean R^2-value on the 5-fold cross-validation can be seen in Table 3.7 below. It shows that the grid search gave different sets of optimal hyperparameters for different values of n. In fact only two models were identical in hyperparameters, the ones with $n = 288$ and $n = 384$.

Table 3.5 Hyperparameters of Model 1-5. Mean test R^2-score on cross validation on the training set, and rank from hyperparameter grid search.

HP / Model	M1	M2	M3	M4	M5
Bootstrap	False	True	False	True	True
F	2	10	2	10	3
S	0.95	0.85	0.3	0.9	1
D	3	3	3	3	5
B	100	100	5000	5000	5000
n	384	384	384	384	384
Mean R^2-score CV	0.883	0.880	0.882	0.880	0.869
Rank	1	94	6	106	2490

Table 3.6 R^2-score of Model 1-5 on the validation and test data set. $n = 384$.

Objective / Model	M1	M2	M3	M4	M5	BM
Validation data	0.9141	0.9213	0.9213	0.9213	0.9070	0.8784
Test data	0.8906	0.8838	0.8906	0.8838	0.8838	0.7813

3.4 Sweep of the Hyperparameter *n*

The Random forest model 3 was fitted and compared to the Baseline model. Both models were given increasing amounts of feature data (n). Fig. 3.10 illustrates the R^2-values calculated from Eq. (2.17). Two models for different values of n are shown. The figure shows a clear gap between the Baseline model and the random forest model when the length n is small, 0 to 60 days the random forest have a distinct advantage over the Baseline model. However, when passing 60 days the Baseline model actually performs better.

Fig. 3.11 illustrates the MANE_n-values calculated from Eq. (2.18). This figure also shows a clear gap between the Baseline model and the Random forest model when the length n is small. Between 0 and 140 days Model 3 is better than the Baseline Model. However, when passing 140 days the Baseline model performs better. It is also interesting to note that the difference in n for RF and BM reaching 0.10 in MANE_n, Model 3 achieves this around 16 days and the Baseline model achieves this around 70 days.

Table 3.7 The hyperparameters of the best models for each n, and their mean test R^2-value from cross validation on the training data.

n	Bootstrapping	F	S	D	B	Mean test R^2
1	True	1	0.900	3	100	0.802
3	False	1	0.950	3	100	0.818
6	False	1	0.500	3	1000	0.826
12	False	1	1	3	100	0.824
24	False	2	0.900	3	700	0.834
48	False	2	0.500	3	700	0.844
72	True	9	0.950	3	700	0.853
96	False	2	0.850	3	400	0.860
144	True	10	1	3	400	0.865
192	True	10	0.950	3	1000	0.868
288	False	2	0.950	3	100	0.874
384	False	2	0.950	3	100	0.883
480	False	2	1	3	100	0.884
720	False	2	0.900	3	400	0.893
960	False	2	0.750	3	1000	0.899
1440	False	3	0.950	3	700	0.916
2160	False	3	0.950	3	400	0.947
2880	False	3	0.900	3	700	0.969
3600	False	4	0.500	3	100	0.981
4800	False	5	0.850	3	100	0.989

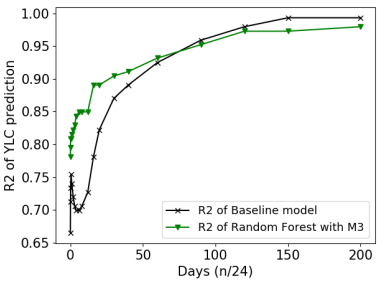


Figure 3.10 R^2-values of RF M3 and Baseline model as a function of n on the test data set.

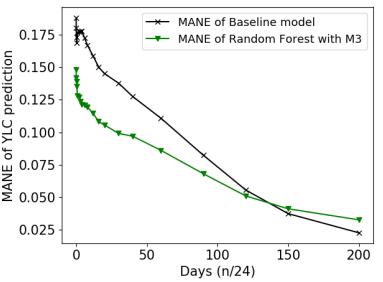


Figure 3.11 MANE-values of RF M3 and Baseline model as a function of n on the test data set.

A flowchart of the resulting model selection process can be seen in Fig. 3.12.

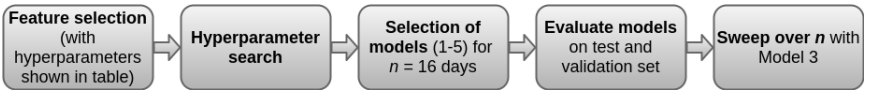


Figure 3.12 Flowchart of the model selection process. Table 3.2 is the table referred to in the figure.

3.5 Modelling Results

The main modelling results with the test data are presented in Fig. 3.14, 3.15 and 3.13. For Fig. 3.14 and 3.15 are the samples sorted since they are histograms and EDFs, it will be easier for the models to accurately predict the distribution rather than individual samples.

The Fig. 3.13, shows predictions of YLC for different n from Random forest model 3 and the Baseline model paired with a sorted list of test target values. This figure shows that the variation in absolute values are larger when there are larger YLC values and vice versa. It shows that studying R^2-values gives a high importance to the values with a large YLC which corresponds with a low YTL (short lifetime). The components with short lifetime are especially interesting to study from a product design perspective. The figure also show that the predictions both of M3 and Baseline model becomes significantly better when larger n is used.

In Fig. 3.16 histograms are shown of the target, the Baseline- and the Random forest predictions. Since they are displayed in a histogram, they have been separately ordered in ascending order. Therefore the histogram is not paired. This type of plot is especially useful for providing the design team with data. It is also useful for testing whether the distribution of the predictions are close to the target distribution. Fig. 3.17 shows the EDF and it is clear that the Random forest model 3 gives a closer prediction to the true distribution than the Baseline model.

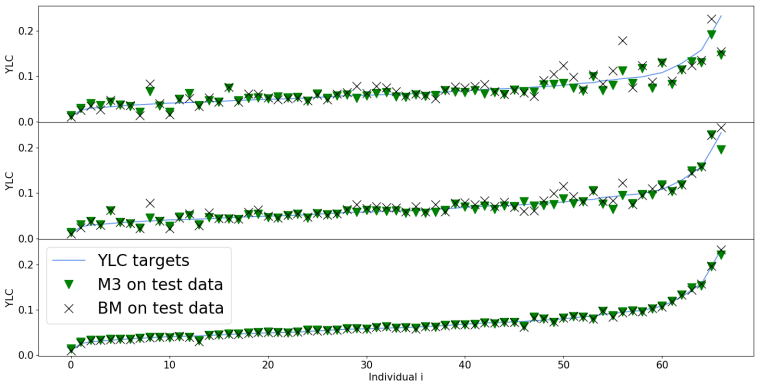


Figure 3.13 Random forest model 3, Baseline and target values of YLC, sorted by target in ascending order on the test data. For the upper, middle and lower graph the amount of data hyperparameter n is 1, 384 (16 days) and 4800 (200 days) respectively. For better visualisation the number of samples have been reduced. After sorting every 7th sample is kept.

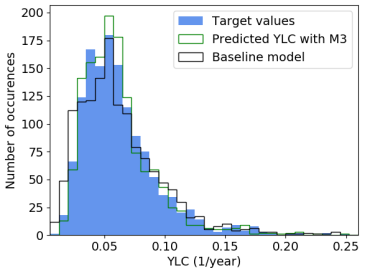


Figure 3.14 Histograms of Baseline and M3 predicted YLC-values on the test set, for n = 384 (16 days).

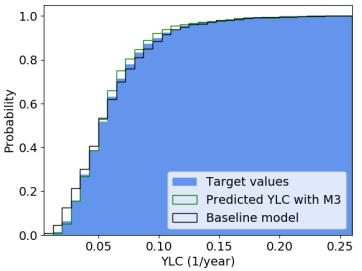


Figure 3.15 Empirical distribution function of predicted YLC-values on the test data set from the Baseline model and M3, for n = 384 (16 days).

Hypothesis Testing

It is of interest to examine how close the Random forest and the Baseline model are to the true target distribution. Here $\hat{G}_n(t)$ are the EDF of YLC given by Random forest model 3 and the Baseline model, where n indicates the number of hours that were used. $\hat{F}_N(t)$ gives the EDF of the YLC target values. The result is presented in Table 3.8 and 3.9. Random forest model 3, have a significant difference of mean but a non significant difference in distribution compared to the target distribution. For

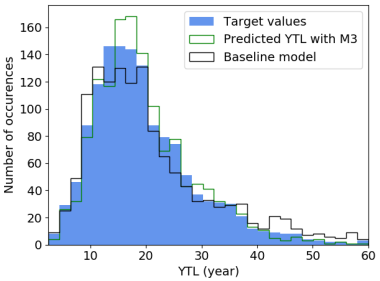


Figure 3.16 Histograms of Baseline and M3 predicted YTL-values on the test set, for $n = 384$ (16 days).

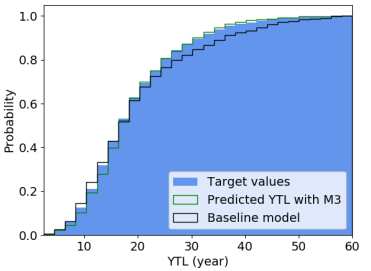


Figure 3.17 Empirical distribution function of predicted YTL-values on the test data set from the Baseline model and M3, for $n = 384$ (16 days).

the Baseline model, it is the opposite, it has a non-significant difference of mean and a significant difference in distribution.

Table 3.8 The paired t-test of same YLC mean hypothesis $H_0 : \mathbb{E}[F(x)] = \mathbb{E}[G(x)]$. The tests are conducted on the test set with $n = 384$ (16 days).

Comparison	t statistic	p-value
M3 vs Target	-6.302	0.000
Baseline vs Target	-1.386	0.166

Table 3.9 The Kolmogorov-Smirnov test of same YLC distribution hypothesis $H_0 : F(x) = G(x)$. The tests are conducted on the test set with $n = 384$ (16 days).

comparison	D_n statistic	p-value
M3 vs Target	0.046	0.111
Baseline vs Target	0.067	0.004

3.6 Predictions on Extended Data Set

In the data set there are 6954 individuals that have temperature time series that are at least $n = 384$ data points long. All of these individuals are put in an extended data set. This data set is fed into Random forest Model 3 and Baseline model to forecast the lifetimes of many cameras. Fig. 3.18 show a histogram with number of individuals in the modelling data set and the extended data set. The number of time series for each product can be seen in Table A.2 in Appendix A.

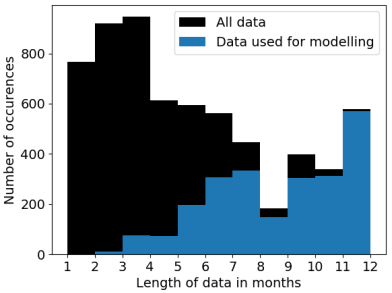


Figure 3.18 Histogram of the amount of data available in months. All individuals have at least $n = 384$ temperature data points.

Histograms and empirical distribution functions of model predictions of YLC and YTL are shown in 3.19, 3.20, 3.21 and 3.22. The predictions of Random forest model 3 and Baseline model are shown. Note that Fig. 3.19 and 3.20 show that Model 3 predict the probability to be smaller for small and large values of YLC, compared to the Baseline model. Model 3 have a larger peak of occurrences around 0.05.

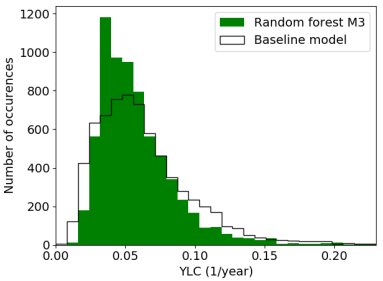


Figure 3.19 Histograms of predicted YLC, from the Baseline model and M3 for $n = 384$ (16 days).

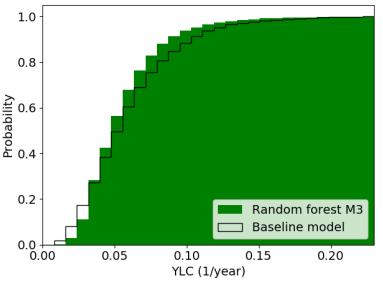


Figure 3.20 Empirical distribution function of predicted YLC, from the Baseline model and M3, for $n = 384$ (16 days).

Fig. 3.21 and 3.22 show the inverted YLC, YTL values, i.e. the number of years a camera is predicted to operate, for the extended data set. The relation between Model 3 and the Baseline model are the same here.

In Table 3.10 descriptive statistics of YTL predictions in are shown. Something interesting to note is that only 0.5 % of the individuals live shorter than 5 years according to M3.

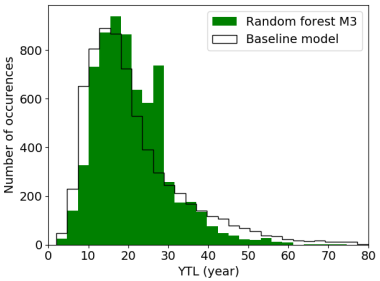


Figure 3.21 Histograms of predicted YTL, from the Baseline model and M3 for $n = 384$ (16 days).

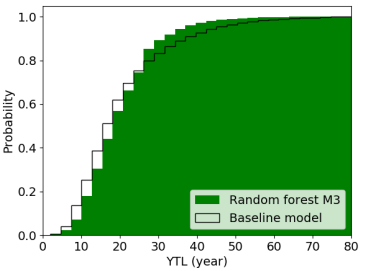


Figure 3.22 Empirical distribution function of predicted YTL, from the Baseline model and M3, for $n = 384$ (16 days).

Table 3.10 Descriptive statistic of predictions on extended. Values shown in YTL (years).

	RF M3	BM
#Samples	6954	6954
Mean	20.9	21.7
Std	9.3	14.0
Min	3.4	2.3
25 %	14.2	12.7
50 %	19.3	17.9
75 %	26.3	26.2
Max	89.7	214
$\mathbb{P}(\text{YTL} \leq 5)$	0.5 %	0.8 %
$\mathbb{P}(\text{YTL} \leq 10)$	7.0 %	13 %
$\mathbb{P}(\text{YTL} \geq 30)$	13 %	19 %

4

Discussions

4.1 Data Quality - Discussion

Jitter

From the results of the jitter experiment, it is clear that the jitter that appears in the signal has an insignificant effect on the YLC and was therefore ignored.

Interpolation techniques

The Naive interpolation performs equally well for all different gap lengths, but worse than Vallier and linear interpolation. The comparison between the interpolation techniques (Fig. 3.1- 3.2) show that the performance of Vallier interpolation is comparable to that of linear interpolation for small (1-6 data points) and large (48 data points) gaps. However, it performed significantly better where the missing data points were in the range of 10-40. Since Vallier interpolation is at least as good or better than the other available techniques, it was chosen.

A possible source of error is that this evaluation was performed solely on one product (M2026-LE-MK II). This product was designed for outdoor weather, which in general leads to a strong periodicity of 24 in the time series. For indoor products, linear interpolation could perform better. In the future, this could be worth investigating.

Even though Vallier interpolation appears to perform well in general, the fraction of missing values in the time series used for the development of the model is very large (see Fig. A.2). Consequently, the errors created by interpolation is considered to be a liability in the models.

Concatenation of time series

From the comparison between the two distributions (Fig. 3.5) it was possible to conclude that the distributions may be the same since only one of the tests (KS-test) rejected the null hypotheses. If the two distributions were the same, this would

indicate that the internal temperature process was unchanged between two time series. This leads to the conclusion that the interruptions did not occur because of the temperature.

It would have been preferable to conduct the same investigation for the interruptions between the time series for the other products and not only for M2026-LE-MK II. Unfortunately, this was not possible at the time of this investigation.

The histogram in the Fig. A.1 shows that the vast majority of the length of the gaps between the time series used for the development of the models was 1-2 hours. As Vallier interpolation shows good performance in this respect, it is determined to be of less importance than the missing values within the time series.

Target validation

As can be seen from Table 3.1, the required number of months to obtain a valid target value varies significantly between the products. Generally, outdoor cameras require a larger number of months. As the temperature outside the cameras heavily impacts the temperature inside the camera, this was quite expected. The temperature in outdoor cameras varies significantly more than cameras operating indoors. This fluctuation affects the YLC of the camera, making it harder to obtain a good estimate from less data.

For some of the products, the number of individuals was quite low, especially for C 360-P and C Dome-WV, where only a single individual from each product was available due to the absence of long temperature time series. Only one time series from each of these two products contained 11 months of temperature data. It can of course be questionable to allow so few individuals for the validation of the target value. As the alternative was to either remove these products from the data set or use less data for validation, for instance, 10 months instead of 11, it was concluded that this would be sufficient. Removing the products from the data set would decrease the diversity in the data set and limiting the data for the validation could influence the accuracy in the estimate as well.

For a few of the products which were included in the data set before removing the invalid targets, there were no individuals with 11 months of data. Therefore, the individuals from these products had to be removed. Further, the reliability of the target values could be questioned as only 11 months of data was available.

Bias in the data

It is clear from Table A.2 in Appendix A that there are imbalances between the data sets. It is imbalanced in two ways;

1. Individuals per product vary between the products.
2. The sample ratio between the number of individuals per product in the two data sets "total" and "total modelling".

For item 1, as can be seen in the "#Total modelling" column of Table A.2 in Appendix A, the product with the largest number of individuals (M3045-V) has 934, and M3046-1-8mm has the smallest number of individuals (25). Therefore, M3045-V is present 37 times more than M3046-1-8mm in the data. Consequently, the models are better at predicting YLC-values of individuals from M3045-V.

For item 2, the largest ratio is given by M2026-LE-MkII and the smallest ratio given by M3048-P with 18 and 1.4 respectively. This yields a bias in the data, it stems in part from the tough requirement set in Table 3.1. M2026-LE-MkII (outdoor) requires 9.97 months of data, while M3048-P (indoor) requires only 2.88 months. Data bias occurs when the available data is not representative of the true population. This is considered a significant problem and source of error. The final model predicts on 1482 individuals from M2026-LE-MkII but only 83 is used for the modelling.

4.2 Random Forest

Feature selection

It seemed reasonable to use values motivated by the literature for the hyperparameters in the feature selection process. The possible influence of the hyperparameters on the feature importance was not studied. It is possible that another subset of features would have been selected if the hyperparameters were changed, as is the case with the length of the time series (n) given to the model. In addition, no analysis was conducted on whether or not 16 days would be the optimal length of the time series, it was just presumed to be a reasonable length.

The features in Table 2.1 was included in the feature selection. From Fig. 3.9, one can observe a very small variation in the model's performance after adding the image sensor size as a feature. The best model was obtained with the use of 10 features. However, using 17 features gave roughly the same results. The feature containing random values (rand) was included as the second to last feature, indicating that it was among the least important features. This indicates that the other features contributed with predictive power for the model.

The feature that gave the best model by itself was YLC_n mean, which was somewhat expected, since it should express the target value's behaviour fairly well. However, YLC_n could be better since it is calculated from all the temperature data given to the model instead of only the mean value. It is also possible that YLC_n max could be better since it is the high temperatures that damage the capacitor most.

Apart from the sequential forward feature selection method, other feature selection methods could have been applied. It is possible that another method would yield a different model, involving different features. A comparison between different methods could contribute to a clearer picture of the importance of the different features.

Furthermore, additional features could have been investigated, such as the processor of the camera, the power consumption during the interval, and the periodicity of temperature within the interval. An investigation of these features is, however, outside the scope of this work.

Hyperparameter optimisation

The range of the hyperparameters for which the grid search was conducted was concluded to be sufficient, even though some hyperparameters were untuned. The reason being that the untuned hyperparameters were closely related to the minimum number of samples at the leaf node (D), as they also affect the pruning of the trees, but in a different manner. A lower D could have been included in the grid search, but due to the risk of overfitting, this was not done.

As an alternative to the grid search, a more computationally efficient method could have been used. However, it was not necessary in this case since computational time was not a limiting factor. Hence, it was possible to evaluate all interesting combinations of hyperparameters, and therefore a different method would not have yielded a better model.

Model selection

It was moderately surprising that a model without bootstrapping gave the best results of the 5 fold cross-validation on the training set (see Table 3.5) for some values of n. The bootstrapping of the training set is supposed to increase the performance of the Random forest algorithm. Because of this surprising result, the highest-ranked model with bootstrapping was selected for further analysis as well (M2).

In addition, models with the highest number of estimators (5000) were of interest, as according to previous studies, it should not harm the model to increase the number of estimators. Furthermore, the accuracy of the model with only literature motivated or default hyperparameters were selected. It could be used as a benchmark of the importance of hyperparameter optimisation for the used data set.

When evaluating the R^2-scores of model 1 to 5 on validation and test set as in Table 3.6, it is clear that Model 3 gave the best results.

Model analysis

When examining the five models' performance based on two weeks of data, what initially became clear was the small difference in performance between the models, even though the hyperparameters' values and the rank of the models varied. Table 3.5, shows that if bootstrapping is unused the number of features considered before each split (F) was set to 2 for the models which gave the best results. This is likely to compensate for removing one of the factors which contributes to the randomness of Random forest. Setting F very low increases the diversity between the trees and decreases the variance, which is necessary, in order for the model not to overfit.

When bootstrapping was used, the best models used the maximum value of F (10), thus removing this contribution to the randomness of Random forest and decreasing the diversity between the trees. However, as both cross-validation was used on the training set, and as the performance was similar on the validation set, the risk of overfitting was established to be low.

Model 3 only used a sub-sample size of 30 % of the training set. This was rather surprising as using a larger percentage would give the model more information, presumably leading to a better model. However, as 5000 estimators were used in this case this probably compensated for the small sub-sample size. Because of the small difference in R^2-score between the models in general, the fact that a size of 30 % achieved better results than 100 % could have been by chance.

When considering the best models for the other values of n (Table 3.7) the relation between bootstrapping the data set and using few features (and vice versa) appears to exist for every model. This strengthens the conclusion above about it being necessary to compensate for removing one of the factors which contribute to the diversity of the trees.

Not a single of the best models used the maximal number of estimators (5000), this could like previously mentioned be by chance as the difference between the models for a certain n is very small. Because of the very small variation in accuracy it is not possible to conclude that including more estimators harms the models.

Model evaluation

As the accuracy between Model 5 and the other models varied only slightly on the validation and test set, one could argue that the importance of hyperparameter optimisation for this data set is minimal.

When comparing Model 3 with the Baseline model for different values of n (Fig. 3.10 and 3.11), it is clear that Model 3 outperforms the Baseline model for small values of n. However, the Baseline model still obtains reasonable R^2-scores for small values of n and outperforms Model 3 for $n \geq 2160$ (90 days). One possible reason why BM outperforms Model 3 in this case is that when n approaches N, YLC_n approaches YLC_N. The Baseline model is based on YLC_n and may be very similar to the targets. That feature is also present in the Random forest. However, the Random forest may have a harder time to reach a R^2-score of 1 fast. This is because the other features may confuse it, even when YLC_n is a very good predictor of YLC_N

Further, the Baseline model received a significantly lower R^2-value on the test set compared to the validation set and compared to the other models when $n = 384$. Although the different data sets have been investigated, the reason for the variation in performance between the data sets is still unknown.

Fig. 3.11 shows a 10 % mean absolute normalised error around 16 days for Model 3 and around 70 days for the Baseline model. Thus, the model managed to reduce

the required length of time series needed to obtain an acceptable error. This was a goal of the work, to develop a model that can predict with an accuracy at or lower than 0.10 in MANE_n. Having n of 16 days compared to 70 days gives a large increase in number of individuals M, later when predicting on the extended data set. This extended data set contain 6954 individuals compared to the original number of 2328 in the modelling data set.

If the data set were to contain more individuals operating outdoors the difference between the models would probably be larger since the large variation in temperature during the year generally makes the Baseline model perform worse. Unfortunately, many of the removed individuals (due to the absence of long time series) were operating outdoors. They would have contributed to a wider diversity in the time series data. Thus, removing these individuals presumably decreased robustness in the models.

The ambition was to use the best Random forest model for each value of n (Table 3.7) when comparing Random forest to the Baseline model for different values of n (Fig. 3.10 - 3.13). However, this turned out to be very time-consuming, and therefore Model 3 was used for all the values of n. Model 3 was selected as it obtained the best R^2-values on both the validation and test set (Table 3.6) when $n = 384$.

A strength of Model 3 and the Baseline model are the absence of a clear error trend as a function of the target values. This is concluded from Fig. 3.13.

Distribution analysis

The Kolmogorov-Smirnov test showed that the YLC distribution predicted by the Random forest model was statistically close to the target distribution. This result shows the predictive power of the Random forest model 3. They were so similar that the Kolmogorov-Smirnov test were unable to differentiate between the two distributions. However, the paired t-test rejected the null hypothesis of same mean. This implies that the mean of the target YLC distribution was not the same as the mean of the Random forest YLC predicted distribution. This result is quite surprising since two distributions that are the same, per definition have the same mean. However, the KS-test is passed because the distributions are statistically the same not because they are exactly the same. That is, Random forest model 3 predictions can pass the KS-test without necessarily passing the t-test.

The Baseline model passed the t-test but not the KS-test. This is perfectly reasonable, two different distributions can have the same mean. It is however, an indicator that the Baseline models distribution is not adequately similar to the target distribution.

For the purpose of this work, the distribution of Random forest predictions is superior to the Baseline distribution as it is of more value to obtain an accurate distribution over an accurate mean.

Concept drift

Unfortunately a non-i.i.d. data was not considered during the development of the models. This could be improved in the future when more data is available. In order to mitigate concept drift, the models should be continuously updated as more data becomes available.

It was decided early in this work that because of the scarcity of data, it was not feasible to use a non-i.i.d. data set. This data set sometimes included target values based on very short time series, which led to unreasonably good predictions, as the amount of data (n) given to the models were about the same length as the time series. Later, when filtering out some of the short time series based on the target validation, it was considered less of a problem. However, the decision to not include a non-i.i.d. data set remained, and is now considered a possibility to improve the models.

Modelling results

If the target distribution of the YLC and the distribution of the Random forest predicted YLC-values are concluded to be the same, one could argue that the Random forest predictions in Fig. 3.19 show the true distribution of the YLC values for the entire camera population. This was shown to be likely in the Kolmogorov-Smirnov test comparing the target- with the RF distribution. However, as the lifetime formula for capacitors was simplified in this work the true distribution will likely be different. How the simplified formula impacts the behaviour of the distribution is unknown and is worth investigating in the future.

5

Concluding Remarks

This work has;

1. Improved the quality of temperature time series by developing and implementing the Vallier interpolation method to impute missing values.
2. Achieved the goal of obtaining a quantitative assessment of how well the yearly lifetime consumption can be predicted.
3. Developed a model for lifetime prediction which requires significantly less data (16 days) compared to the Baseline model (70 days), for a mean absolute normalised error of 10 %.
4. Concluded that hyperparameter optimisation for Random forest regression does not improve the performance for this data set.
5. Obtained a distribution of the lifetime for the entire camera population, which could be used to guide future design.

6

Populärvetenskaplig Sammanfattning

Det är av största vikt att kondensatorerna som sitter i Axis övervakningskameror håller lagom länge. Om vi har kunskap om livslängden för kondensatorer med olika kvalité så kan vi balansera behovet för bra kvalité med att spara pengar. I detta arbete utvecklas matematiska modeller som använder temperaturdata som samlas in i realtid. Dessa används för att förutspå hur länge kondensatorer kommer att fungera. Genom att använda datadrivna modeller kan vi veta med större säkerhet hur länge komponenterna håller.

För att lösa detta problem utvecklas en basmodell och flera Random forest-modeller. Basmodellen används som jämförelse och Random forest-modellerna utvecklas för att bli så bra som möjligt. Det visar sig att både basmodellen och den bästa Random forest-modellen kan väl förutspå livslängden för kondensatorerna. Eftersom det finns begränsningar i mängden data för många kameror är vi också intresserade av att förutspå livslängden till en viss noggrannhet med så lite data som möjligt. Om det går att noggrant förutspå livslängden med en liten mängd data kan vi inkludera maximalt antal kamera individer och på så sätt få bättre förståelse vilken kvalité som är nödvändig. Detta klarar Random forest-modellen av betydligt bättre än basmodellen. Med en sänkning från 70 till 16 dagar är den mer avancerade modellen en stor vinst.

Arbetet kan komma att användas i kommande produktutveckling på Axis. Med det presenterade resultatet skulle elektronikingenjörer kunna göra mer fakta grundade beslut vilket kan gynna både Axis som företag och dess kunder.

Något överraskande var mängden problem som uppstod med temperaturdatan. Från början saknades en stor del (cirka 30 %). Detta problem åtgärdades genom att interpolera saknade värden. Olika interpolations tekniker utvecklas och testas, därefter tillämpas den bästa.

Tillgängligt finns vissa sanna värden på kondensatorernas livslängd. Trots det fanns det stora problem med att veta vilka av dessa värden som var pålitliga. Metoder för att studera detta utvecklas och jämförs. När vi kommit fram till en bra metod används den för att försäkra oss om vilka värden som vi kan och inte kan lita på.

A

Appendix

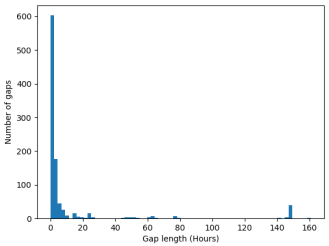


Figure A.1 A histogram over the lengths of the different gaps of missing data (total 1000 gaps) of the analysed time series from the product M2026-LE-Mk II. Note that the majority of the gaps have a length under five hours.

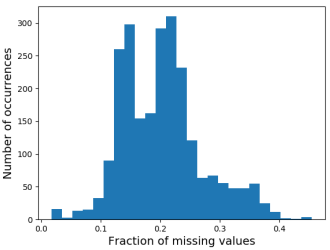


Figure A.2 A histogram of the fraction of missing values in all of the time series used in the development of model ('Total modelling' in table A.2).

(a) C Dome V. *(b) M3045-V.* *(c) M2026LEMkII.*

(d) M3046-V. *(e) M3044-V.* *(f) M3106-LVE-MkII.*

(g) C 360 P. *(h) M3047-P.*

Figure A.3 Sample pictures of products. Product names given in the figure.

(a) P3245-LVE. *(b) M2026-LE.* *(c) M3048-P.*

(d) M4206-LV. *(e) C Dome WV.*

Figure A.4 Sample pictures of products. Product names given in the figure.

Table A.1 Capacitor component constants, specific to each product. The constants are defined in section 2.1.

Product	L_0	T_0	C	T_{offset}
C Dome V	2000	105	10	3
M3045-V	2000	105	10	3
M2026-LE-MkII	2000	105	10	3
M3046-V	2000	105	10	3
M3044-V	2000	105	10	3
M3106-LVE-MkII	2000	105	10	3
C 360 P	2000	105	9	3
M3047-P	2000	105	10	3
P3245-LVE	1000	105	9	3
M2026-LE	2000	105	10	3
M3048-P	2000	105	10	3
M3106-L-MkII	2000	105	10	3
P1365 Mk II	2000	105	10	3
M3106-LVE	2000	105	10	3
M4206-LV	2000	105	8	3
C Dome WV	2000	105	10	3
M3046-1-8mm	2000	105	10	3

Table A.2 Number of individuals from each product in the different data sets used for the development of the Random forest models. The total number of available time series with a length of at least 16 days is also included in the table under column #Total. The products with 0 individuals was first included but later removed due to lack of time series with 11 months of data.

Product	#Total	#Total modelling	#Train	#Validation	#Test
C Dome V	0	0	0	0	0
M3045-V	1897	934	593	151	190
M2026-LE-MkII	1482	83	53	10	20
M3046-V	773	554	349	96	109
M3044-V	653	156	103	27	26
M3106-LVE-MkII	609	41	24	7	10
C 360 P	391	153	98	24	31
M3047-P	357	133	89	22	22
P3245-LVE	0	0	0	0	0
M2026-LE	271	48	34	6	8
M3048-P	244	176	114	21	41
M3106-L-MkII	0	0	0	0	0
P1365 Mk II	0	0	0	0	0
M3106-LVE	0	0	0	0	0
M4206-LV	0	0	0	0	0
C Dome WV	141	25	18	4	3
M3046-1-8mm	136	25	14	5	6
Total	6954	2328	1489	373	466

Bibliography

Aha, D. W. and R. L. Bankert (1996). "A comparative evaluation of sequential feature selection algorithms". In: *Learning from data*. Springer, pp. 199–206.

Albertsen, A. (2010). "Electrolytic capacitor lifetime estimation". *Bodos Power Magazine*, pp. 52–54.

Bernard, S., L. Heutte, and S. Adam (2009). "Influence of hyperparameters on random forest accuracy". In: *International Workshop on Multiple Classifier Systems*. Springer, pp. 171–180.

Bocock, G. (n.d.). *Electrolytic capacitor lifetime in power supplies*. URL: https://www.xppower.com/resources/blog/electrolytic-capacitor-lifetime-in-power-supplies.

Box, G. E. and D. R. Cox (1964). "An analysis of transformations". *Journal of the Royal Statistical Society: Series B (Methodological)* **26**:2, pp. 211–243.

Breiman, L. (2001). "Random forests". *Machine learning* **45**:1, pp. 5–32.

Brownlee, J. (2018). *A gentle introduction to statistical power and power analysis in python*. https://machinelearningmastery.com/statistical-power-and-power-analysis-in-python/. (Accessed on 05/02/2020).

Cherry, B., B. V. Kumar, and S. Yaduvir (2018). "Condition monitoring of aluminium electrolytic capacitors using accelerated life testing: a comparison". *International Journal of Quality & Reliability Management* **35**:8, pp. 1671–1682. ISSN: 0265-671X. DOI: 10.1108/IJQRM-06-2017-0115. URL: https://doi.org/10.1108/IJQRM-06-2017-0115.

"Arrhenius' Equation" (2008). In: Chesworth, W. (Ed.). *Encyclopedia of Soil Science*. Springer Netherlands, Dordrecht, pp. 49–49. ISBN: 978-1-4020-3995-9. DOI: 10.1007/978-1-4020-3995-9_38. URL: https://doi.org/10.1007/978-1-4020-3995-9_38.

Cutler, A., D. R. Cutler, and J. R. Stevens (2012). "Random forests". In: *Ensemble machine learning*. Springer, pp. 157–175.

Dıaz-Uriarte, R. and S. A. De Andres (2006). "Gene selection and classification of microarray data using random forest". *BMC bioinformatics* **7**:1, p. 3.

Fernández-Delgado, M., E. Cernadas, S. Barro, and D. Amorim (2014). "Do we need hundreds of classifiers to solve real world classification problems?" *The journal of machine learning research* **15**:1, pp. 3133–3181.

Friedman, J., T. Hastie, and R. Tibshirani (2001). "The elements of statistical learning". In: vol. 1. 10. Springer series in statistics New York, pp. 587–605.

Gupta, A., O. P. Yadav, D. DeVoto, and J. Major (2018). "A review of degradation behavior and modeling of capacitors". In: *International Electronic Packaging Technical Conference and Exhibition*. Vol. 51920. American Society of Mechanical Engineers, V001T04A004.

Guyon, I. and A. Elisseeff (2003). "An introduction to variable and feature selection". *Journal of machine learning research* **3**:Mar, pp. 1157–1182.

Hodges, J. L. (1958). "The significance probability of the smirnov two-sample test". *Arkiv för Matematik* **3**, pp. 469–486.

Hutter, F., H. Hoos, and K. Leyton-Brown (2014). "An efficient approach for assessing hyperparameter importance". In: *International conference on machine learning*, pp. 754–762.

Jakobsson, A. (2013). *An Introduction to Time Series Modeling*. Studentlitteratur.

Jones, D. R., M. Schonlau, and W. J. Welch (1998). "Efficient global optimization of expensive black-box functions". *Journal of Global optimization* **13**:4, pp. 455–492.

Kohavi, R., G. H. John, et al. (1997). "Wrappers for feature subset selection". *Artificial intelligence* **97**:1-2, pp. 273–324.

Parler, S. G. and P. Dubilier (2004). "Deriving life multipliers for electrolytic capacitors". *IEEE Power Electronics Society Newsletter* **16**:1, pp. 11–12.

Probst, P., B. Bischl, and A.-L. Boulesteix (2018). "Tunability: importance of hyperparameters of machine learning algorithms". *arXiv preprint arXiv:1802.09596*.

Probst, P. and A.-L. Boulesteix (2017). "To tune or not to tune the number of trees in random forest". *The Journal of Machine Learning Research* **18**:1, pp. 6673–6690.

Probst, P., M. N. Wright, and A.-L. Boulesteix (2019). "Hyperparameters and tuning strategies for random forest". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **9**:3, e1301.

Sankaran, V. A., F. L. Rees, and C. S. Avant (1997). "Electrolytic capacitor life testing and prediction". In: *IAS '97. Conference Record of the 1997 IEEE Industry Applications Conference Thirty-Second IAS Annual Meeting*. Vol. 2, 1058–1065 vol.2.

Seaman, S., J. Galati, D. Jackson, and J. Carlin (2013). "What is meant by" missing at random"?" *Statistical Science*, pp. 257–268.

Segal, M. R. (2004). “Machine learning benchmarks and random forest regression”. *escholarship*.

unknown (2020). “Nonparametric statistics and model selection”. *MIT* **http://www.mit.edu/ 6.s085/notes/lecture5.pdf**, pp. 1–3.