Institute of Architecture of Application Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

An analysis of four server power models

Franz Sebastian Müller

Course of Study:	Softwaretechnik
Examiner:	Prof. Dr. Marco Aiello
Supervisor:	Brian Setz, M.Sc. Stephan Urbanski
Commenced:	April 1, 2022
Completed:	Oktober 3, 2022

Abstract

In cloud computing research, many models are used to predict server power. However, a lot of these models are not sufficiently tested on industry hardware due to lack of access to this type of hardware in academia. In this work, we address this need for model evaluation and lack of data from real data centers.

We obtain a large dataset from a data center of the company AEB SE, located in their headquarters in Stuttgart. The dataset contains hardware utilisation data on the averages of CPU-frequency, CPU utilisation, server power consumption, and ambient temperature as well as peak power consumption. These metrics are measured at five-minute intervals over the span of a year, for all 73 servers. We use the information on average CPU utilisation and average server power to train four server power models that use CPU utilisation to predict the power consumption of a given server. Two of these power models are from literature, and the other two are our own work. We form server groups, based on a combination of hardware characteristics the servers have, such as CPU models, server types and storage sizes. We then train the models on them and compare the accuracy the models have. This answers the question which hardware characteristics should be considered when grouping servers as a basis for training the power models and which distinctions are unnecessary.

We also compare the models to each other, based on their accuracy, generalisability and speed of training. We find that one of our models was in all but a few cases the most accurate one. It also generalises better than the other three models and is one of the two fastest models in training. However, it does have the issue of predicting inaccurate and sometimes even semantically incorrect results in higher CPU utilisation areas.

In plotting the server power samples at specific CPU utilisations, we observe that the general shape of these plots resembles a horizontal asymptote. Therefore we propose a model that tries to imitate this general shape.

Unfortunately, the dataset we obtain is heavily biased towards lower CPU utilisation areas, which may introduce an error in our evaluation of the two least accurate models, one of ours and one from literature, both of which are dependent on using power measurements obtained at full utilisation.

The dataset we obtain is freely available for research and can be used to evaluate other power models, or in other research that requires hardware utilisation data from a data center.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

The research on how to schedule tasks in data centers in an energy efficient way is of great interest right now because the rising demand for cloud services in the industry leads to ever increasing sizes of these facilities [DWF16]. The more hardware is deployed, the more energy is consumed and high energy consumption is a concern both from an economical as well as from an ecological point of view.

But in developing scheduling algorithms to diminish this consumption, data center researchers face the problem of generally not being able to test their architectures and algorithms on industry hardware because academia usually does not have the resources needed for building a data center which is comparable to industry standard [BVWS14]. Thus, researchers have to rely on models that tell them how much energy a data center server consumes in a production environment under a given load as a substitute for real-time measurements.

There are many kinds of power models for a data center server, some taking just CPU utilisation as their sole input, some taking other components into account, like networking, cooling, storage and so on [IM20].

Generally, the lack of data gathered in industry data centers is a problem for evaluating the accuracy of these models [BVWS14]. Many cloud companies are reluctant to grant access to their facilities and apart from some researchers like Fan et al. [FWB07] or Radovanovic et al. [Rad+22], who are allowed to have access, cooperation between academia and industry in this area is lacking [BVWS14]. This leaves researchers no choice but to test their ideas on small testbeds, that cannot be an adequate substitution for hardware data from an industry size data center [BVWS14].

We are fortunate to get access to a data center at AEB SE for this paper and can obtain data points on Average Power, CPU utilisation, Peak Power, Ambient Temperature and CPU Frequency, sampled every five minutes, for a period of one year on 73 s ervers. The dataset that we gather here is a contribution that addresses the need for real world data and will be published as part of this thesis.

We take two models from literature that reportedly have a high accuracy and evaluate them on this large dataset to find out if their reported accuracy can be r eplicated. We also introduce two power models of our own here. All models have parameters that need to be fitted to some measurements before the models can be used.

We train the models and compare them to each other with the metrics R-squared and Root Mean Squared Error (RMSE). We will elaborate further on our methodology in Chapter 5. For a definition of these metrics, please see Section 2.2.

Some server types are present with different storage sizes, while having otherwise identical hardware characteristics. We will answer the question whether or not these differences in storage size may be disregarded, when categorising servers into datasets for power model training. None of the power models we evaluate take storage into account but have a single input variable, namely CPU

utilisation. If storage size can be disregarded, without losing accuracy, this would make the power models more generalisable. It would also make them easier to use in practice because retraining the models whenever a server with a different amount of storage is installed would not be necessary anymore.

Similarly, there are some server types that share a CPU model but are otherwise different. We compare the accuracy of the power models when disregarding all differences in the other hardware of the server types and combining them based on their CPU model. We do this to answer the question if the distinctions based on CPU models might be enough. Finally, we evaluate the power models for the speed at which they can be trained.

The structure of our paper is as follows:

First, we will outline some background information necessary for understanding this paper in Chapter 2. Next, we are going to give an overview of related work in the field of server power models in Chapter 3. Thereafter, we are going to present the approach that we take to gather and evaluate the data in chapter three. Subsequently in Chapter 6, we present the evaluation of the trained models and the dataset that we collect. The results of this Chapter are then discussed in Chapter 7. Finally in Chapter 8 we will provide our outlook for future work to be done and conclude.

In the next Chapter, some relevant background information on technologies and metrics that we use is introduced.

2 Background Information

To make our work more accessible to readers from a broad range of backgrounds, we will now provide information on technologies and concepts which are important to understand our work.

2.1 Integrated Lights-Out and OneView

In the data center of AEB SE, all servers are manufactured by the Hewlett Packard Enterprise (HP). This company installs a sensor and remote control unit called Integrated Lights-Out (iLO) [Ilo] on every recent HP server. This device may be used to remotely power the device on or off, check server health status, or gather utilisation data, like we are doing in this paper. [Ilo]

The iLO data is aggregated in the OneView Application [Oneb]. This application provides, among other things, a dashboard, where Server Health status can be checked [Oneb]. It also provides a REST API, where we call a specific Endpoint [End] to retrieve the utilisation data for an individual server.

2.2 Metrics

To assess how accurately the power models predict, we use two metrics, which we will introduce in the next two sections.

2.2.1 R-squared

As a definition of the R-squared value, the authors of [MPP19] write: “In statistics, the coefficient of determination R^2 is the proportion of the variance in the dependant variable that is predicted from the independent variable(s).”. This metric can be used to compare how well the four power models can predict the power consumption of a server at a given CPU utilisation. As Malkina-Pykh and Pykh [MPP19] note, there are multiple possible ways to calculate an R-squared value that do not necessarily lead to the same result. We use two functions of the python library scikit to calculate the R-squared value, namely the score [Scie] function of the Linear regression module and the r2_score function [Scic], which both calculate the R-squared value as follows:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.1}$$

Here, y_i are the actual values, while $\hat{y}_i$ are the corresponding predictions of the model and $\bar{y}$ is the average across all actual values.

In order to assess if a model is a good fit or not, we follow the approach Zhang et al. [ZLQZ13] took and consider a model a good fit that has an R-squared value of 0.95 or higher. However, as Malkina-Pykh and Pykh [MPP19] point out, the R-squared value is not suitable for models that are non-linear in their parameters. This concerns two of the power models we evaluate here. Hence, we include another metric as well, the root mean squared error. For further discussion on the inadequacy of the R-squared value, please see Section 7.6 on page 60.

2.3 Root Mean Squared Error

The Root Mean Squared Error is, as the name suggests, the Square Root of the Mean Squared Error. We again use a utility function from the libray scikit, this time called mean _squared _error, which, according to the documentation [Scib], calculates the mean _squared _error like this:

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2 \tag{2.2}$$

We then take the square root of the result to obtain the root mean squared error (RMSE).

$$RMSE(y, \hat{y}) = \sqrt{MSE(y, \hat{y})} \tag{2.3}$$

This metric represents the average distance the predictions of a model have from the actual values, disregarding wether that distance is due to overprediction or underprediction. This metric is in the same unit as the underlying prediction, so watts in the case of the power models we evaluate.

It is important to note that in analysing the values of this metric, they are comparable only on the same dataset. For example, when a model has an RMSE value of 40 watts, that can be quite a large error if the underlying servers consume a maximum of 120 watts, but it can be a relatively small error if the servers consume up to 1500 watts. So the power models can only be compared based on their RMSE values, if these values originated from training and testing them on the same server configuration.

2.4 Repeated Kfold cross-validation

In machine learning it is common that models have parameters which are fitted to a portion of the dataset, called the training-sample. The remainder of the dataset is then used to assess how well the model can predict values that it was not trained on. This is called the test sample. Depending on how this splitting into training and test samples is done, a bias can be introduced to how the model parameters are calibrated. For example, all data points in the training samples might be in the lower quartil of the dataset, which could introduce a large error when predicting data from the

upper quartil. This problem may be avoided by using a repeated kfold cross-validation approach. The dataset is randomly split into n samples and n-1 of those are used to train the models, while the remaining one is used for testing it [Man20]. All n splits become the test sample once [Man20]. This process of splitting and evaluating on each of the splits is then repeated m times. Depending on the number of splits and repeats, this approach minimises the bias introduced by the splitting choice as well.

2.5 SPEC power benchmark

When researchers cannot obtain data on the power consumption a server type has at a given CPU utilisation, they can consult the SPEC benchmark [Spea]. The Standard Performance Evaluation Corporation (SPEC) was founded "to establish, maintain and endorse standardised benchmarks and tools to evaluate performance and energy efficiency for the newest generation of computing system" [Spea]. Their power benchmarks go from zero to 100 percent CPU utilisation for a given server type in ten percent steps and report, among other things, the power consumed by the server [Speb].

3 Related Work

When considering the field of server power modeling, we find several other works which evaluate different server power models in literature. Some of these models take into account multiple hardware parameters, such as storage size, CPU frequency or network utilisation [IM20]. Others just take in CPU utilisation, like the models we evaluate in our work.

In a survey, Ismail and Materwala [IM20] provide an extensive overview of different server types, classify them with a taxonomy and evaluate them for their accuracy on a unified test bed, consisting of three different server architectures as well as using two SPEC datasets from different servers. Ismail and Materwala [IM20] evaluate 24 software-based power models in total, using different benchmarking tools to generate load on hardware components, such as network, storage or CPU. This article [IM20] provides a good overview of software-power models and compares them in a unified setup but has the drawback of evaluating the models only on three different server architectures, using benchmarking tools to generate resource utilisation. The two SPEC Power benchmarks are added for models that only take in CPU utilisation.

Whether the examined models will perform in a similar manner on different server architectures or under real load found in data centers remains unclear. So the size of the dataset that Ismail and Materwala [IM20] used, as well as its origin, being benchmarking data instead load from real users, are key differences to our work.

A second, big survey by Dayarathna et al. [DWF16] lists over 200 power models and classifies them according to a taxonomy. Dayarathna et al. [DWF16] not only consider software based server power models but also hardware based models, models for virtual machine power consumption and models of larger scale systems, such as data center power models. The authors of [DWF16] compare the models to each other but do not perform any experimental evaluation themselves. This is a key difference to our work because we train and evaluate all models that we discuss on data from a data center. Dayarathna et al. [DWF16] also do not propose a model of their own, whereas we propose two new models. Dayarathna et al. [DWF16] provide a good overview of the field, which was the aim of their review.

Möbius et al. [MDS14] conducted a survey, in which they list 20 power models, on subcomponent, virtual machine and entire server level. Möbius et al. [MDS14] compare seven server power models and also highlight the different direct measurement techniques available at the time. The authors of [MDS14] did not compare the models on their own experimental setup but rather cite the estimation errors from the original papers.

McCullough et al. [McC+11] evaluate five different power models with the use of various benchmarking tools on an Intel Calpella platform. McCullough et al. [McC+11] compare three models using lasso regression, one using support vector machines and the MANTIS model proposed

by Economou et al. [ERKR06]. McCullough et al. [McC+11] conclude that while the models performed with low single digit error rates on full system power prediction, they have high mean error rates of up to 150 percent in predicting subcomponent power consumption.

We also review some smaller surveys smaller surveys, such as the one conducted by Rivoire et al. [RRK08], in which they train and evaluate five different server power models, four from literature and one which they propose themselves in this work. The authors of [RRK08] evaluate the five models on five different hardware platforms, using benchmarks to stress different hardware components. There is one major drawback to the work of Rivoire et al. [RRK08]: they misrepresent one of the models they evaluate, namely the non-linear model of Fan et al. [FWB07], which we are actually evaluating as well in this work. The authors of [RRK08] evaluate the wrong equation here, making their observations on its accuracy much less valuable.

In the next section, we will introduce the work of Fan et al. [FWB07] in more detail, as their non-linear equation is one of the two equations from literature we will evaluate alongside our works.

3.1 "Power Provisioning for a Warehouse-sized Computer" by Fan et al. [FWB07]

Fan et al. [FWB07] are Google employees and as such are able to obtain access to large scale industry data centers at Google LLC. In the study "Power Provisioning for a Warehouse-sized Computer"[FWB07], the authors analyse power usage data from 15,000 servers, and consider three types of load, namely from the mail-service Gmail, the Google Websearch service and from Map Reduce Jobs on a subset of 5,000 servers.

The authors of [FWB07] evaluate the size of the gap between the theoretically possible peak power consumption of a data center and the actual peak power that ends up being consumed. Fan et al. [FWB07] also discuss a few methods to optimise this utilisation of data center facilities in a manner that minimises the likelihood of outages, such as power capping or CPU voltage/frequency scaling.

$$P(u) = P_{idle} + (P_{busy} - P_{idle}) \times u \tag{3.1}$$

$$P(u) = P_{idle} + (P_{busy} - P_{idle}) \times (2 \times u - u^r) \tag{3.2}$$

Relevant to our study is mainly that Fan et al. [FWB07] propose two server power models, one linear, the other having a parameter in the exponent that needs to be fitted to a training dataset. The two equations are shown as equation 3.1 and 3.2. P_{idle} denotes the power drawn by a server around zero percent CPU utilisation and P_{busy} at around 100 percent CPU utilisation, while u denotes the CPU utilisation in percent (though expressed as a value between zero and one) [FWB07]. Fan et al. [FWB07] compare the two models and come to the conclusion that the non-linear model fits the data better. The authors of [FWB07] aggregate the predictions of these models to a Power

Distribution Unit level (several hundreds of servers) in order to dynamically predict power usage. At this aggregated level, Fan et al. [FWB07] find that the non-linear model has a mean error of below one percent provided that a fixed offset is removed.

Especially the non-linear model of Fan et al. [FWB07] received alot of attention in subsequent works by other researchers. In their evaluation of software-based server power models, Ismail and Materwala [IM20] compared the non-linear model 3.2 to ten other single variable CPU-based power models, finding it to have an error of around twelve percent both for the servers of the unified setup that Ismail and Materwala [IM20] used, as well as for the SPEC power benchmark datasets. In comparison to the other ten single variable CPU-based power models Ismail and Materwala [IM20] ranked it 6th and 8th most accurate respective to the unified setup and the SPEC dataset. When compared over all benchmarking tools that Ismail and Materwala [IM20] used, and compared to all 24 other software-based server power models that they evaluated, Ismail and Materwala [IM20] found the non-linear model 3.2 of Fan et al. [FWB07] to be only the 18th most accurate.

Möbius et al. [MDS14] included the non-linear equation 3.2 in their survey and concluded that it “stands out in terms of reproducible results and portability” [MDS14]. However, they do not perform an experimental evaluation as a part of their survey.

Dayarathna et al. [DWF16] also list the non-linear equation 3.2 as part of their study but do not evaluate it further. The authors of [DWF16] call this work and especially the linear equation 3.1 “highly influential in recent server power modeling research”. As already mentioned, Rivoire et al. [RRK08] include both the linear model 3.1 as well as the non-linear equation 3.2, but only the evaluation of the linear equation 3.1 is valid, as they unfortunately misrepresent the non-linear equation 3.2.

In another study [Rad+22] where the authors have access to Google data centers, Radovanovic et al. [Rad+22] compare the linear model 3.1 to two models that they propose here, showing that the newer models outperform 3.1. Radovanovic et al. [Rad+22] refer to the linear model 3.1 as the “benchmark model” [Rad+22] and state that data center operations at Google used this model for power prediction previously, but now at Google it is replaced by newer models.

Due to this interest in these equations and the impact the paper of Fan et al. [FWB07] had, we pick the non-linear equation 3.2 as one of the models we evaluate further in this paper. Model 3.2 sometimes predicted very accurately, like in [FWB07] but sometimes less accurately [IM20] than other comparable models. This makes another evaluation useful, as this might provide further insight into how accurately this model 3.2 predicts server-power.

3.2 “A high-level energy consumption model for heterogeneous data centers” by Zhang et al. [ZLQZ13]

The other model we evaluate here is a work of Zhang et al. [ZLQZ13], from their paper: “A high-level energy consumption model for heterogeneous data centers”. In this paper [ZLQZ13] evaluate 3 models on 392 SPEC power benchmark results. Zhang et al. [ZLQZ13] also perform benchmarking experiments on two servers themselves. Zhang et al. [ZLQZ13] propose linear,

quadratic and cubic equations, finding that the cubic one 3.3 outperforms the other two. The authors find that their cubic model performs with an R-squared value greater than 0.98 on all SPEC power benchmarking results they consider.

The cubic model 3.3 of Zhang et al. [ZLQZ13], which we evaluate further here, is of the form:

$$P(u) = a + b \times u + c \times u^2 + d \times u^3 \tag{3.3}$$

u is again the CPU utilisation in percent, this time not in the range of 0 to 1, but given as integers though. All other variables need to be fitted to a dataset in the training process.

We choose this model for our evaluation because its reported accuracy is high. Ismail and Materwala [IM20] compared ten other single variable CPU-based server power models to this one and found it to be second most accurate, having an error of estimation of below five percent. When compared to all other 23 models they considered, the cubic model 3.3 of Zhang et al. [ZLQZ13] ranked 13th most accurate [IM20]. Ismail and Materwala [IM20] also evaluate the other two models proposed by Zhang et al. [ZLQZ13] and they confirm their relative accuracy to each other, which Zhang et al. [ZLQZ13] reported, with the cubic model being the most accurate and the linear model the least accurate.

4 Models

A major contribution of this paper is the introduction of two new CPU utilisation based server power models. In this chapter, we will discuss their structure as well as our thought process in their creation.

4.1 Polynomial Model

Since we evaluate the polynomial model 3.3 of Zhang et al. [ZLQZ13] in this paper and one of their major findings was, that their cubic model outperformed their quadratic and linear models [ZLQZ13], we want to inquire whether or not this trend continues to an even higher power. Therefore, we introduce another power model of the form:

$$P(u) = a + b \times u + c \times u^2 + d \times u^3 + e \times u^4 + f \times u^5 \tag{4.1}$$

This new model also needs to be trained to obtain fitting parameters a - f to the server being modeled. Identically to 3.3 the CPU utilisation u is in percent here, expressed in integer numbers.

4.2 Asymptotic Model

After an initial analysis, we noticed that the general trend of plotting the server power consumption at a CPU utilisation generally does not have the form of exponential curve, or of odd powered polynomials but rather that of a horizontal asymptote.

There is a sharp increase in power consumption in the CPU utilisation range of zero to twenty percent and afterwards the plots flatten off. This is true for figures 7.2, 6.8a, 7.7 as well as in the ones we include in Appendix B. Because of this general observation, we introduce a server power model that captures this trend of the plots.

$$P(u) = a + \frac{P_{idle}}{(u + c)^2} \tag{4.2}$$

Here, P_{idle} is the power that the servers consume when they are at below two percent CPU utilisation. We average the values of all power consumption samples. All datasets we consider have such low utilisation samples.

As an initial guess for a, we give the server-power consumption at full CPU utilisation. Whenever we cannot average samples out of the dataset for this value because the server was never utilised to that extent, we use the *calibrated maximum power* value as reported by the OneView API [Onea] instead. As we discuss in Section 7.5 on page 56, this might introduce an error to power model 3.2 of Fan et al. [FWB07] but also to our asymptotic model. For c we find that an initial guess of 0.5 delivered good results.

The parameters are then callibrated to the dataset at hand and the training process is closer to model 3.2 of Fan et al. [FWB07], than to that of model 3.3 of Zhang et al. [ZLQZ13] and 4.1 because we use the python function of curve_fit [Scid] in the training of 3.2 of Fan et al. [FWB07] as well.

5 Methodology

5.1 Data Gathering

As already mentioned, for this work we gather data from a data center at AEB. We got permission from AEB to query the OneView API, through which we obtain utilisation data like ambient temperature in celsius, CPU frequency in Mega Hertz, CPU utilisation in percent, peak power in watts, and average power in watts. All metrics are measured in a five-minute interval and averaged except in the case of peak power, where iLO records the highest value only. All data was returned in a JSON format schematically visible in Figure 6.3. We can retrieve a dataset of three days per request and write a Java Spring Application that repeatedly calls the API Endpoint [End] and stores each three-day sample in a Mongo Database until we retrieve the full dataset going back one year for all servers.

The OneView Documentation states that there should be data on the same metrics being averaged over a one-hour interval as well [End]. This one-hour average should be stored for up to three years [End]. Unfortunately, this feature did not seem to be available in the servers at AEB.

5.2 Training of the models

All four power models we consider have parameters that need to be fitted to a training dataset and only then can be used for prediction. For the model 3.3 of Zhang et al. [ZLQZ13] and our polynomial model 4.1, we use the python library scikit [Scia] with its Polynomial Features and Linear Regression modules [Scif]. For the two other models, 3.2 of Fan et al. [FWB07] and our asymptotic model 4.2, we use the curve fit function [Scid] of the python library scikit.

Here, initial guesses for the parameters need to be provided. For the parameter r of the model 3.2 of Fan et al. [FWB07], we provide the value of 1.4 that they mention in their paper [FWB07]. For our asymptotic model, the initial guesses are the maximum power consumption for parameter a and a value of 0.5 for parameter c, which we find brings about the most accurate results.

Model 3.2 of Fan et al. [FWB07] also needs the power consumption at full CPU utilisation as a constant in the equation. When the servers are ever utilised to that extent, we take the average of all average power samples that have a corresponding CPU utilisation of above 97 percent. When such values are not present, however, which is, unfortunately, the rule and not the exception, we take a value of *calibrated maximum power* as reported by the OneView API [Onea] instead. How close this value is to the actual maximum power consumption that we observe in some servers and what effect the usage of this value might have on the accuracy of models 3.2 of Fan et al. [FWB07] and model 4.2, is a subject we will discuss further in Section 7.5 on page 56.

All models are trained on the dataset using a repeated kfold cross-validation approach with five splits, repeated ten times. So in total, we train and and test the power models on each configuration fifty times. Every time we calculate the metrics discussed in 2.2 and take the average of them as the value we report in our evaluation. We did this using the RepeatedKFold [Rep] module of the sklearn library.

6 Evaluation

In this chapter, we will perform a general analysis of the dataset and then show how we train the four power models with different subsets of the dataset. We call these subsets server configurations or simply configurations. What we mean by this is a server, or a group of servers, that share certain hardware characteristics. We categorise the servers into configurations, based on their server type (e.g. BL460c), storage size, or CPU model (e.g. Intel Xeon E5-2640).

We do this to better understand which hardware characteristics are important to consider in clustering the servers for power model training and which can be disregarded.

First, we will discuss the structure of the data center of whic we obtain the dataset in Section 6.1. Then we are going to make a general analysis of the dataset in Section 6.2. In Section 6.3 and 6.4, we compare the accuracy of the power models on configurations that disregard storage size differences, to their accuracy on those that do not. In Sections 6.5.1 and 6.5.2, we will determine whether or not it is possible to combine different server types that have the same CPU model into one configuration for training, or if that leads to too large a loss in power model accuracy.

6.1 Data center structure

At AEB the data center is deployed on two floors of their headquarters in Stuttgart. Both floors have uninterruptible power supply in the form of a battery backup and a diesel generator that is used in case of longer outages. They use both blade servers and normal rack servers, and the heat generated by the servers is partially used to heat the headquarters of AEB. AEB handles storage externally from the servers, so most of them only have minimal storage capacity installed.

6.2 General analysis of the data

We obtain data from 73 different servers, all produced by HP. In total, we collected 7,604,756 five-minute samples, 7,308,761 of which we analyse further, after filtering out the null values. We display a histogram of the CPU utilisation of the entire dataset in Figure Figure 6.1. Note that the y-axis is in millions.

In analysing the histogram, we observe that almost half of all data points are in the range of zero to ten percent CPU utilisation, with another smaller peak in the range of forty to sixty percent. The dataset is very sparse when it comes to CPU utilisations above 90 percent. We will discuss this characteristic of the dataset further in Section 7.4. The configurations vary in size. Some configurations have only a single server that has these hardware characteristics. The largest configuration, where no

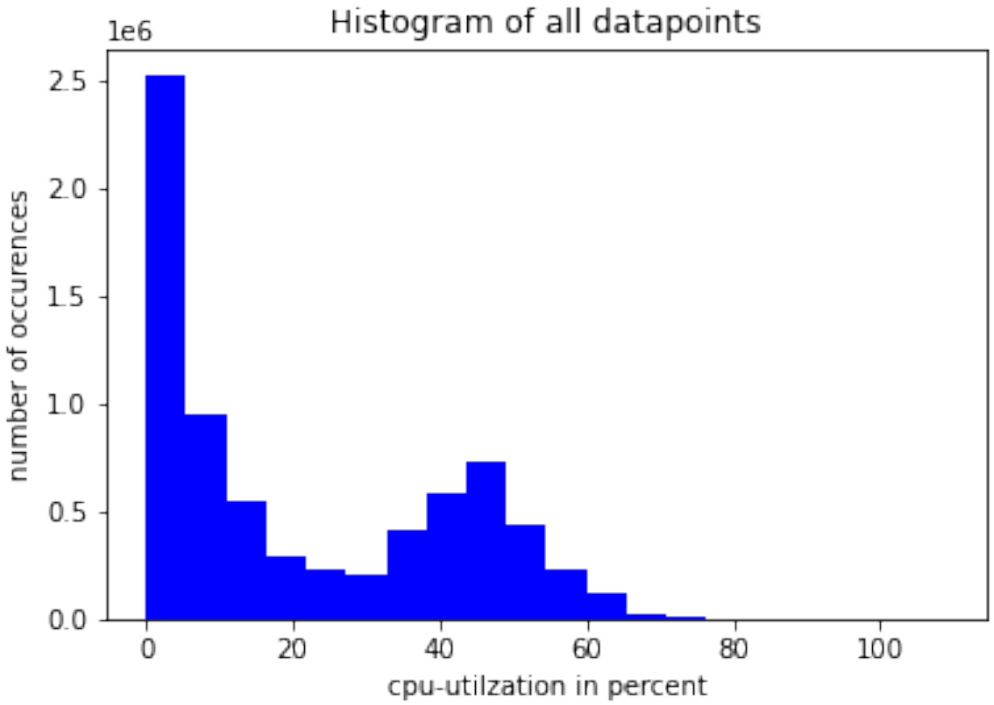


Figure 6.1: Histogram of all data points

hardware distinctions are made, has 73 servers in it. From an individual server we obtain around 100,000 data points. The oldest server type we obtain data from is the ProLiant BL460c, an HP blade server of the eighth generation.

In Figure 6.2, we display the number of occurrences (in millions) of a given power consumption. These come from 73 servers, measured in 5-minute intervals over one year. We observe that the range of power consumed by the servers reaches from around 50 watts to around 600 watts. The bulk of the power consumed by the servers across all data points lies between 200 and 350 watts, with well over four million data points lying in that range.

Figure 6.3 showcases the general format of he JSON Response delivered by the OneView API. We leave out the metadata to highlight the structure of the actual metrics we analyse. Each data point has a timestamp - given as the number of milliseconds since midnight January first, 1970 - and the value of the metric. As mentioned, one such response contains an array of metrics, which are each sampled every five minutes. The OneView API returns in one response the equivalent of around three days of samples. This time range can then be adjusted by the parameters *sliceStartTime* and *sliceEndTime*, which the script we use for data collection does automatically for us. The time range between *newestSampleTime* and *oldestSample* Time is the range in which samples are currently stored on the server. Whenever new samples are added, this time range shifts, and the oldest samples are deleted.

The *metricCapacity* property, which we leave out due to space restrictions in all but the *AmbientTemperature* metric, displays the maximum value the metric took in this three-day interval. The metric *PowerCap* is always null in our dataset because AEB does not use a power capping scheme.

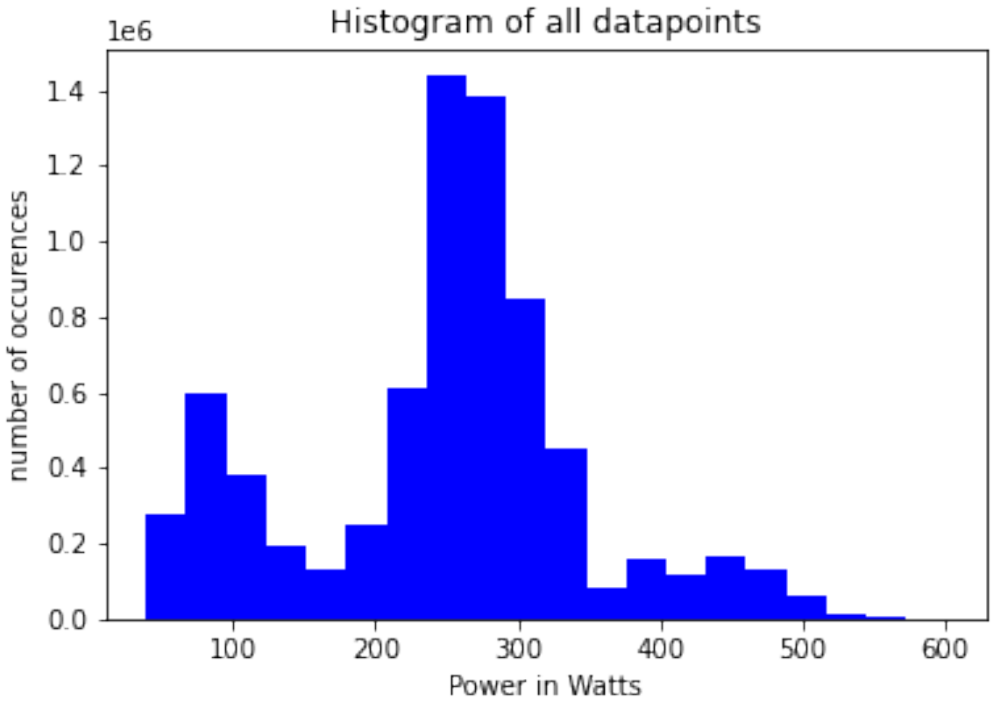


Figure 6.2: Histogram of power consumption

Since AEB deploys several layers of virtualisation in their operations, it is unfortunately not possible for us to link any of the hardware utilisations we observe here to business processes. The processes are automatically shifted to the machines that have free resources and it is not possible for us to reconstruct this behaviour for the historical dataset we obtain from the OneView API.

We train all four power models on different portions of the data, to test their accuracy as well as their generalisability. First, we focus on server types that are present as servers with different amounts of storage. We aggregate these servers and train the power models on this aggregation in the next Section. In Section 6.4, we then split these servers by their storage size and retrain the power models on each partition. This is all to see if storage size consideration makes a difference in their accuracy or if these differences can be disregarded.

6.3 Models without storage consideration

Some servers at AEB have the same server type and CPU model but are present with different amounts of storage installed on them. The main persistence of data is handled in dedicated units by AEB, not in the servers we analyse here. Nonetheless, the comparatively small amounts of storage installed on the servers vary.

Since all four power models that we evaluate do not consider storage, we first train them on every server type, without consideration for how much storage is installed on the individual servers. In Section 6.4, we consider this difference and compare the accuracy of the power models.

Figure 6.3: General Schema of the JSON Response

To avoid introducing a bias, we do consider every other difference in hardware, such as server type, CPU model, number of cores, et cetera, as its own configuration for training. This allows us to answer the question whether or not it is possible to disregard the storage sizes in training the power models, without losing too much accuracy of prediction.

Server models that are only present in the data center with one amount of storage installed are excluded as well here. Therefore, we only display the five server configurations here, for which there are servers that have different storage sizes. These results can be seen in table Table 6.1.

server-type	cpu-model	cpu-freq	core-count	server-count	r-sq	rmse	power-model
BL460c	X.E5-2660v3	2.6GHz	10	7	0.969	9.13	Polynomial
BL460c	X.E5-2660v3	2.6GHz	10	7	0.930	13.62	Zhang et al
BL460c	X.E5-2660v3	2.6GHz	10	7	0.609	32.28	Fan et al
BL460c	X.E5-2660v3	2.6GHz	10	7	0.907	15.7	Asympt.
BL460c	X.E5-2667v4	3.2GHz	8	6	0.936	12.86	Zhang et al
BL460c	X.E5-2667v4	3.2GHz	8	6	0.956	10.65	Polynomial
BL460c	X.E5-2667v4	3.2GHz	8	6	0.902	15.84	Asympt.
BL460c	X.E5-2667v4	3.2GHz	8	6	0.197	45.47	Fan et al
BL460C	X.E5-2640	2.50 GHz	6	3	0.941	8.95	Polynomial
BL460C	X.E5-2640	2.50 GHz	6	3	0.887	12.34	Zhang et al
BL460C	X.E5-2640	2.50 GHz	6	3	0.687	20.59	Fan et al
BL460C	X.E5-2640	2.50 GHz	6	3	0.870	13.25	Asympt.
SY480	X.G.6132	2.6GHz	14	21	0.866	11.84	Zhang et al
SY480	X.G.6132	2.6GHz	14	21	0.945	7.61	Polynomial
SY480	X.G.6132	2.6GHz	14	21	0.032	31.81	Fan et al
SY480	X.G.6132	2.6GHz	14	21	0.711	17.39	Asympt.
SY480	X.G.6248	3GHz	24	15	0.996	6.68	Polynomial
SY480	X.G.6248	3GHz	24	15	0.993	8.23	Zhang et al
SY480	X.G.6248	3GHz	24	15	0.961	19.8	Fan et al
SY480	X.G.6248	3GHz	24	15	0.957	20.77	Asympt.

Table 6.1: Server configurations split by server type, CPU model, frequency and core-count, disregarding storage

We train all four power models on each server configuration, making their relative performance easily comparable. We observe that our polynomial power model 4.1 predicts most accurately in every one of these five cases - achieving an R-squared value that is always above ninety-three percent - with the best value being 0.996. The RMSE reflects this accuracy as well, with power model 4.1 predicting with the least average distance to the actual value in all the five server configurations.

Power model 3.3 of Zhang et al. [ZLQZ13] predicts very accurately as well. It has R-squared values higher than 0.87 in all cases, with one almost perfect score of 0.99. With these values, power model 3.3 of Zhang et al. [ZLQZ13] is the second most accurate of the four models in the five configurations of Table 6.1. The RMSE values confirm this relatively high accuracy.

Our asymptotic power model 4.2 performs reasonably well in all cases as well, with the R-squared values ranging between 0.87 and 0.96. The highest RMSE value is 20.77 here. These results make it the third most accurate model in these configurations.

The predictions of model 3.2 of Fan et al. [FWB07] lack accuracy in many server configurations with one R-squared value as low as 0.03. It is important to note here that the R-squared value is, as further discussed in Section 7.6 on page 60, not a good metric to use when comparing models that are non-linear in their parameters. However, the RMSE values of 3.2 and the plotting of the predicted graphs also indicate a big discrepancy between the predictions of 3.2 and the actual values. This inadequacy of the R-squared metric is important to keep in mind for the remainder of the evaluation, but we will not mention it every time we discuss the R-squared metric going forward. We display the predicted values of all four power models (blue) as well as the true values (red) for two of the five server configurations in Figure 6.6 and 6.7.

Figure 6.6 is particularly interesting as it showcases that all power models except our asymptotic model 4.2 do not inherently follow the asymptotic shape that the power curve has here. They just try to fit the dataset as accurately as possible instead. This leads to better metric results but also to sometimes semantically incorrect results. Our polynomial power model as seen in Subfigure 6.6c and the power model 3.3 of Zhang et al. [ZLQZ13] as seen in Subfigure 6.6d both perform well as far as the metrics are concerned (an R-squared value of 0.96 and 0.94 respectively), but their curves predict a power consumption that takes a sharp, almost exponential, increase the closer the server comes to one-hundred percent utilisation. This does not lead to a high error in this case since the dataset is very sparse in the range of over sixty percent utilisation.

This sparseness of the dataset can be observed in Figure 6.4. We notice that for the majority of the samples, the server is idle. As already indicated, this is a trend that is consistent with the general CPU utilisation distribution of the entire dataset across all servers, which we display in Figure 6.1. We will discuss what effect this sparseness has on the accuracy of the power models in Section 7.4.

Our polynomial power model 4.1 predicts that the power consumption will decrease from forty percent utilisation to around eighty percent utilisation. After that point, it predicts a sharp increase in server-power consumption when approaching full CPU utilisation. That the power would decrease for a period while utilisation increases is not consistent with any of the power curves we observe (the ones that we do not include in the paper are displayed in Appendix B). This semantically incorrect prediction may be due to the dataset being sparse in higher utilisation ranges.

Our asymptotic power model 4.2, however, does follow the general shape of the power curve, leading to predictions that are at least semantically valid throughout the utilisation range. So although this power model has, on average, a higher error than model 3.3 of Zhang et al. [ZLQZ13] and our polynomial model 4.1, it delivers sensible results in the face of a sparse dataset. We will analyse the performance of all power models on sparse datasets more in Section 7.4. Although the shapes of the predicted power curve and the general trend of the samples are similar to that of our asymptotic power model, it overpredicts the consumed power in this server configuration by roughly fifty watts, diverging at around twenty percent utilisation. This still leads to a relatively low accuracy as expressed in the metrics, with an R-squared value of 0.902 and an RMSE of 15.84.

The power model 3.2 of Fan et al. [FWB07] does not predict the first, sharp increase of power consumption in the range of zero to fifteen percent of CPU utilisation, a curve shape that is very common for the server-power curves we display in this paper, as well as in those included in Appendix B. Rather, the power model of Fan et al. [FWB07] predicts a roughly linear increase of power consumption until around the eighty percent utilisation mark, after which the predicted

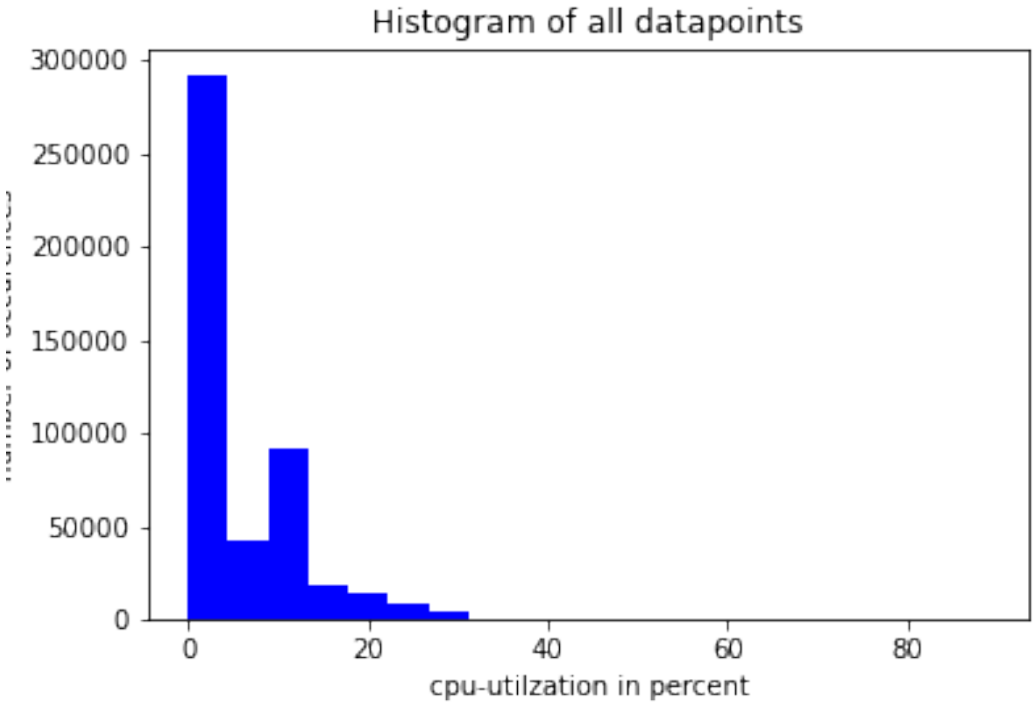


Figure 6.4: Histogram of CPU utilisation samples in BL460c gen9, CPU X.E5-2667, without storage consideration.

power declines. This misses the general shape of the curve for most of the curve and also delivers a semantically incorrect result with the predicted power consumption decreasing as CPU utilisation increases in the range of eighty to ninety percent utilisation.

We also include the plots of the power models trained on the server type SY480 in the configuration where Xeon Gold 6248 is installed as the CPU. We do this because it is the server configuration of the five out of 6.1, where all equations performed best. We can see that the tendency of 3.2 to disregard the initial spike in power consumption and subsequently predict an almost linear curve fits the dataset better here.

The observation that both our polynomial model 4.1 as well as the model 3.3 of Zhang et al. [ZLQZ13] tend to predict a sharp increase - or, in this case, decrease - in power consumption when approaching full CPU utilisation can once more be seen here, with model 3.3 of Zhang et al. [ZLQZ13] predicting diminishing power draw near maximum utilisation, while our polynomial model 4.1 predicts a sharp increase in power drawn when nearing maximum utilisation. This tendency might be due to their shared nature of being odd polynomials.

Our asymptotic model 4.2 performs reasonably well but predicts the increase of power consumption to flatten off while nearing one-hundred percent utilisation, which it does not in this case, or at least, not as strongly as predicted. Figure 6.7 corresponds to the last four entries in Table 6.1.

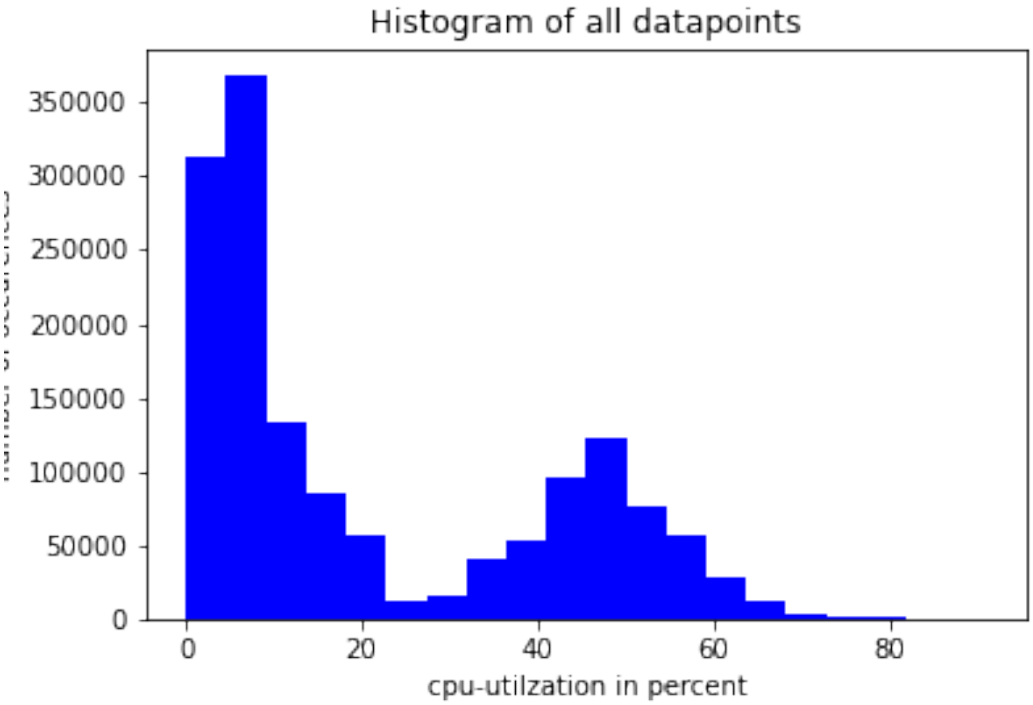


Figure 6.5: Histogram of CPU utilisation samples in SY480 CPU model X.Gold 6248, without storage consideration.

We include the histogram 6.5 of this server configuration here as well because with fifteen servers it is one of the largest server configurations we analyse and its dataset is less sparse in high CPU utilisations than other configurations as seen for example in the histograms 6.4 or 7.3.

In the next section, we will compare the accuracy of the four power models when trained on the same server types as in Table 6.1 but this time making a distinction each time this same server type is present with a different amount of storage installed on the server.

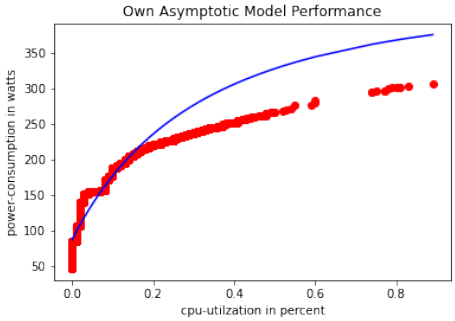
6.4 Power models trained with storage consideration

Four of the five server configurations we analyse in the last section are present with two different storage sizes, while one, BL460c with CPU model Xeon E5-2640, has three different storage sizes. Storage sizes range from 1536 gigabytes to 32 gigabytes. We now will treat each different storage size on each server type as a different server configuration for training and evaluation. In total, this amounts to eleven configurations. In Table 6.2 we list all the accuracy metrics resulting from training the power models on these configurations. The configurations are ordered first by server type and then by storage size.

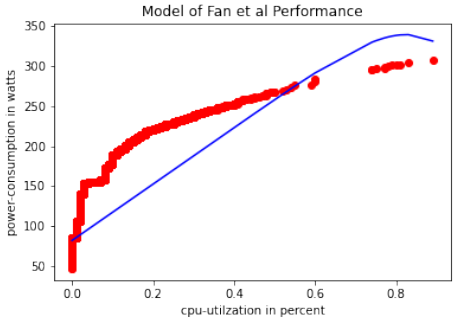
We will now compare the accuracy that the power models have in Table 6.2 to each other as well as to the accuracy they have on the five configurations of Section 6.3.

server-type	cpu-model	cpu-freq	core-count	storage	server-count	r-sq	rmse	power-model
BL460c	X.E5-2640	2.50GHz	6	128GB	1	0.997	2.33	Asympt.
BL460c	X.E5-2640	2.50GHz	6	128GB	1	0.968	7.37	Fan et al.
BL460c	X.E5-2640	2.50GHz	6	128GB	1	0.998	1.91	Polynomial
BL460c	X.E5-2640	2.50GHz	6	128GB	1	0.997	2.39	Zhang et al.
BL460c	X.E5-2640	2.50GHz	6	64GB	1	0.739	5.9	Fan et al.
BL460c	X.E5-2640	2.50GHz	6	64GB	1	0.875	4.08	Polynomial
BL460c	X.E5-2640	2.50GHz	6	64GB	1	0.867	4.22	Zhang et al.
BL460c	X.E5-2640	2.50GHz	6	64GB	1	0.808	5.07	Asympt.
BL460c	X.E5-2640	2.50GHz	6	48GB	1	0.905	0.87	Asympt.
BL460c	X.E5-2640	2.50GHz	6	48GB	1	0.900	0.89	Fan et al.
BL460c	X.E5-2640	2.50GHz	6	48GB	1	0.907	0.86	Polynomial
BL460c	X.E5-2640	2.50GHz	6	48GB	1	0.952	0.62	Zhang et al.
BL460c	X.E5-2660v3	2.6GHz	10	1024GB	6	0.961	4.69	Asympt.
BL460c	X.E5-2660v3	2.6GHz	10	1024GB	6	0.011	23.45	Fan et al.
BL460c	X.E5-2660v3	2.6GHz	10	1024GB	6	0.970	4.1	Zhang et al.
BL460c	X.E5-2660v3	2.6GHz	10	1024GB	6	0.993	1.94	Polynomial
BL460c	X.E5-2660v3	2.6GHz	10	512GB	1	0.871	17.23	Asympt.
BL460c	X.E5-2660v3	2.6GHz	10	512GB	1	−0.395	56.64	Fan et al.
BL460c	X.E5-2660v3	2.6GHz	10	512GB	1	0.997	2.49	Polynomial
BL460c	X.E5-2660v3	2.6GHz	10	512GB	1	0.973	7.83	Zhang et al.
BL460c	X.E5-2667v4	3.2GHz	8	768GB	3	0.908	11.2	Asympt.
BL460c	X.E5-2667v4	3.2GHz	8	768GB	3	0.751	18.46	Fan et al.
BL460c	X.E5-2667v4	3.2GHz	8	768GB	3	0.947	8.49	Polynomial
BL460c	X.E5-2667v4	3.2GHz	8	768GB	3	0.929	9.85	Zhang et al.
BL460c	X.E5-2667v4	3.2GHz	8	384GB	3	0.851	5.28	Asympt.
BL460c	X.E5-2667v4	3.2GHz	8	384GB	3	0.566	9.01	Fan et al.
BL460c	X.E5-2667v4	3.2GHz	8	384GB	3	0.929	3.65	Polynomial
BL460c	X.E5-2667v4	3.2GHz	8	384GB	3	0.916	3.97	Zhang et al.
SY480	X.G.6132	2.6GHz	14	1536GB	8	0.893	12.26	Asympt.
SY480	X.G.6132	2.6GHz	14	1536GB	8	0.183	33.87	Fan et al.
SY480	X.G.6132	2.6GHz	14	1536GB	8	0.977	5.65	Polynomial
SY480	X.G.6132	2.6GHz	14	1536GB	8	0.951	8.26	Zhang et al.
SY480	X.G.6132	2.6GHz	14	1024GB	13	0.925	7.48	Asympt.
SY480	X.G.6132	2.6GHz	14	1024GB	13	0.920	7.71	Fan et al.
SY480	X.G.6132	2.6GHz	14	1024GB	13	0.978	4.06	Polynomial
SY480	X.G.6132	2.6GHz	14	1024GB	13	0.972	4.57	Zhang et al.
SY480	X.G.6248	3GHz	24	1536GB	5	0.954	8.51	Asympt.
SY480	X.G.6248	3GHz	24	1536GB	5	0.953	8.59	Fan et al.
SY480	X.G.6248	3GHz	24	1536GB	5	0.999	1.49	Polynomial
SY480	X.G.6248	3GHz	24	1536GB	5	0.997	2.2	Zhang et al.
SY480	X.G.6248	3GHz	24	512GB	10	0.984	1.88	Asympt.
SY480	X.G.6248	3GHz	24	512GB	10	0.976	2.32	Fan et al.
SY480	X.G.6248	3GHz	24	512GB	10	0.990	1.46	Polynomial
SY480	X.G.6248	3GHz	24	512GB	10	0.990	1.49	Zhang et al.

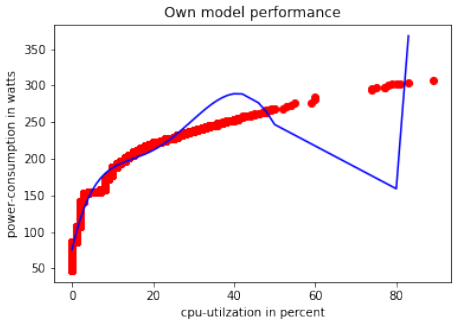
Table 6.2: server types split by CPU model, frequency and core-count, considering storage.



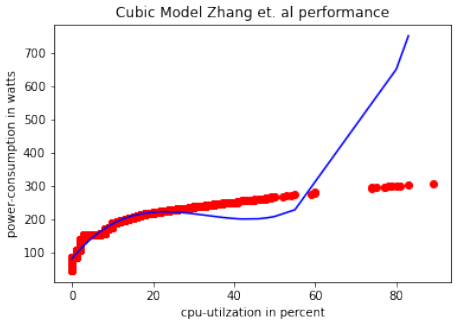
(a) Asymptotic power model



(b) Power model of Fan et al.



(c) Our Polynomial power model.



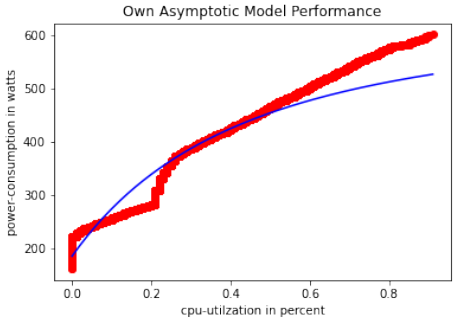
(d) Power model 3.3 of Zhang et al. [ZLQZ13]

Figure 6.6: Power model Performance on BL460c gen 9, Xeon E5-2667 v4 3.2 GHz, 8 core, no storage size consideration.

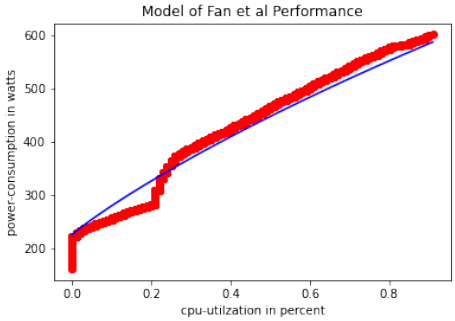
6.4.1 Our Polynomial power model 4.1

Our polynomial power model 4.1 once again predicts very accurately, with five R-squared values being equal to or above 0.99 and with the best value being 0.999. In the server configuration where it is least accurate, it has an R-squared value of 0.875 but still is the most accurate of the power models in this case. It is the most accurate power model when both the R-squared value and the RMSE value are considered in all but one case, where the model 3.3 of Zhang et al. [ZLQZ13] was the most accurate.

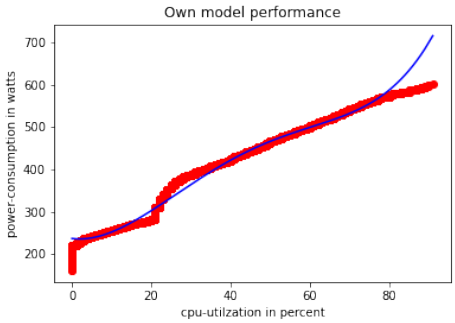
In total, when we compare these results to the accuracy without storage consideration, the accuracy of our polynomial power model 4.1 improves unambiguously in six of the eleven configurations. Unambiguously here means, that the R-squared value increases while the RMSE decreases. In the remaining five configurations, however, the R-squared value decreases, while the RMSE value simultaneously increases. This indicates that power model 4.1 can explain less of the variance in the power consumption, while at the same time predicting with a smaller error. We discuss how this seemingly contradictory development can arise in Section 6.4.5. We note that there is no unambiguously less accurate prediction in these eleven configurations for our polynomial power model 4.1.



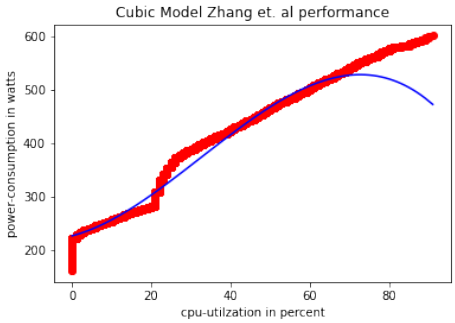
(a) Asymptotic power model.



(b) Power model of Fan et al. [FWB07]



(c) Our Polynomial power model.



(d) Power model 3.3 of Zhang et al. [ZLQZ13]

Figure 6.7: Power model Performance on SY480 gen 10, Xeon Gold 6248 v4 3 GHz, 24 core, no storage size consideration.

6.4.2 Power model 3.3 of Zhang et al. [ZLQZ13]

The power model 3.3 of Zhang et al. [ZLQZ13] has an R-squared value of 0.99 or greater in three of the eleven configurations that we analyse here. In the configuration, where it predicts most accurately, it has an R-squared value of 0.997 and 0.867 when it is least accurate. It is the most accurate power model in one case and comes very close to our polynomial power model 4.1 in most other cases.

When we compare the accuracy 3.3 has here to its accuracy when no storage is considered, it improves in seven out of eleven configurations, while having a decreased R-squared value with a simultaneously decreased RMSE value in the remaining four cases. It does not perform less accurately in any of the eleven configurations when compared to being trained without storage consideration.

6.4.3 Asymptotic power model 4.2

Our asymptotic power model has an R-squared value of greater or equal to 0.99 in one case and when we also consider the RMSE values, the asymptotic power model is the second most accurate power model in this configuration. In all other configurations, it predicts less accurately than our

polynomial power model, and the power model 3.3 of Zhang et al. [ZLQZ13] but more accurately than the power model 3.2 of Fan et al. [FWB07]. When the prediction of the asymptotic power model is least accurate, the R-squared metric is at 0.808.

When compared to training our asymptotic model 4.2 without storage consideration, it improves its accuracy in seven cases. In 3 configurations, we observe an ambiguous result again, with both the R-squared value and the RMSE value decreasing. One configuration leads to an unambiguously less accurate prediction of the asymptotic model - namely the BL460c with CPU model Xeon E5-2660v3 with 512 gigabyte of storage installed.

6.4.4 Power model 3.2 of Fan et al. [FWB07]

The power model 3.2 of Fan et al. [FWB07] predicts in these configurations with a R-squared value between 0.976 and -0.395. We already mentioned the inadequacies of the R-squared metric here, which we will further discuss in Section 7.6, but the RMSE value is also the highest here, with a value of 56.64 watts.

Indeed, model 3.2 of Fan et al. [FWB07] predicts least accurately in all eleven cases, when comparing both the RMSE and the R-squared value to the other power models. However, the differences to our asymptotic power model 4.2 are only in the ranges of the second or third digit behind the decimal point in several cases.

When comparing the results of training model 3.2 of Fan et al. [FWB07] on these eleven configurations with storage consideration, to training it on the same servers but without storage consideration, we observe that the accuracy of prediction unambiguously improves in eight cases. In two cases the already mentioned ambiguous result with a lower RMSE value but also a lower R-squared value appears. In the remaining case, the accuracy unambiguously decreased, with the RMSE increasing and the R-squared value decreasing.

Why would the metric values develop in these apparently contradictory ways? This is a question we want to explore in the next subsection.

6.4.5 Ambiguous Metric Results

We will analyse now the ambiguous results where the power models have a lower R-squared value while also having a lower RMSE value. This seems counter-intuitive because when the error (the RMSE) goes down, we expect that the R-squared value should go up.

To explain this, it is important to remember the definition of the R-squared metric, which we referenced in Section 2.2. In our case, this means that the R-squared value is a metric which determines which proportion of the variance of the server-power consumption (in watts) can be predicted by the changes in CPU utilisation. This proportion changes when the way this relationship of CPU utilisation to power consumption is expressed changes (that is to say, a different power model is used). But it also changes when the proportion of variance that can be predicted by changes in CPU-utilisation in the dataset itself changes.

Therefore the R-squared value decreases in some of the power models trained on a specific configuration because the variance that cannot be explained by the power models using CPU utilisation alone increases. The RMSE decreases as well because the distance of the prediction to the actual value decreases. So the overall error of prediction, the RMSE, decreases here, while the proportion of the variance explainable by the power models also decreases.

This behaviour of the metrics is best illustrated by the following Figure 6.8 of a configuration in which we observe this result for our polynomial model 4.1. In Subfigure 6.8b, we display our polynomial power model 4.1 trained on the server type BL460c with Xeon E5-2640 and 48 gigabytes of storage installed. In Subfigure 6.8a it is the same server type, but this time combining all three different storage sizes that servers of this server type have.

The dataset in Subfigure 6.8b is notably more sparse than the one in 6.8a. This can be seen by the gap in between data points and the fact that the largest data point is at a utilisation of around twenty-five percent. The largest value for power consumption is 85 watts, while the maximum power drawn in 6.8a is around 250 watts.

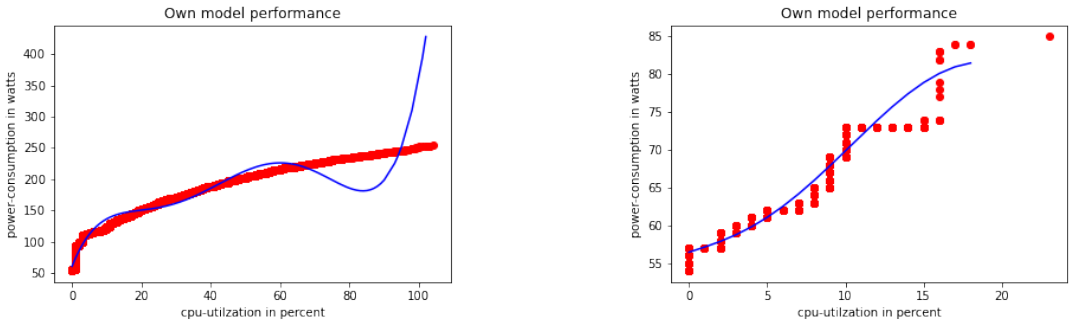
We can explain the difference in RMSE by the example of Figure 6.8 as well. In Figure 6.8a, our polynomial power model 4.1 first predicts a value that is at a maximum around 50 watts below the actual value, in the range from 60 percent utilisation to around 90 percent utilisation. Then the prediction crosses the actual power curve with a steep incline and predicts at its maximum around 150 watts more than was drawn by the servers.

These huge distances have a considerable effect when we calculate the average distance in the RMSE. Meanwhile, in 6.8b, the power consumed by the servers is much lower in the dataset and the power model has a maximum difference between prediction and actual value of around seven watts. This explains the decrease in RMSE when comparing the accuracy of the power model in 6.8a with the one in 6.8b.

The difference in the R-squared value can likewise be explained here, simply by observing how the data points are plotted in 6.8. In 6.8a, the power curve does not seem to have a large variation when compared to the scope of the power being consumed.

In Figure 6.8b, we see more variance that does not correlate well with CPU utilisation. From ten to fifteen percent utilisation, the amount of power drawn remains constant, only to sharply increase by 10 watts near 16 percent utilisation. On the same utilisation percentage (which are integers and individually visible here due to the small scale), some data points vary in power consumption by five to ten watts. These variations are also present in 6.8a but masked by the large scale of the power consumption and the bigger amount of data points (data from three servers instead of only one), so these variations make a smaller proportional difference.

These are the explanations for the seemingly contradictory development of the two metrics. The answer to the question of whether or not the power model is a better fit in 6.8a or in 6.8b is still ambiguous though. In 6.8b there is a larger percentage of variance that cannot be predicted by the power model but the overall error is smaller. In 6.8a a larger proportion of the variance can be predicted by the power model, but the overall error is larger, especially when approaching full utilisation.



(a) Our Polynomial power model without storage consideration.

(b) Our Polynomial power model with 48GB of storage considered.

Figure 6.8: Illustration of simultaneous decrease of R-squared and RMSE on BL460c gen8.

6.5 Combination based on CPU Model

In categorising the different servers into configurations, we notice that there are several different server types that have the same CPU model installed on them. For example, we observe server types DL380 and ML350, which both have the CPU model Xeon E5-2620 version three.

In the next two sections, we first train the server power models on the individual server types and then on a combination based on CPU model. If training the models on such a combination would not lead to a loss in accuracy, then the power models would me more generalisable. They would require no retraining for a new server type with a CPU model, which they were already trained on.

6.5.1 Server models treated as individual configurations

In total, we find four pairs of server types that have the same CPU model installed. We first display in Table 6.3 how accurate the power models are when they are trained on each server type individually. In the case of servers with different amounts of storage installed, we combine them into one configuration here, to avoid introducing a bias. The two servers BL460c with CPU Xeon E5-2640 and SY480 with CPU Xeon Gold 6132 fall into this category - we already analysed them in Table 6.1 on page 31.

When we analyse the metrics, we observe that our polynomial power model is once again the most accurate in both metrics in seven out of eight configurations. In the one in which it is not the most accurate, it is the least accurate with an unsensible, negative R-squared of over 190 watts, which, if we recall the power histogram 6.2, is an incredibly large discrepancy. We display this result in Figure 7.2. The most accurate R-squared metric for our polynomial power model 4.1 is 0.996 with an RMSE value of 2.41 watts.

However, while being generally more accurate than the other three models, our polynomial power model 4.1 has an R-squared value of greater than 0.95 in only three out of eight cases. The threshold of 0.95 is the value after which the prediction of a model can be considered useful.

The power model 3.3 of Zhang et al. [ZLQZ13] is once again the second most accurate power model in all but the case in which our polynomial model 4.1 also predicted less accurately. In the configuration in which it predicts most accurately, it has an R-squared value of 0.981 and an RMSE value of 5.15 watts. This is the same configuration that 4.1 also performs the most accurately in. The two power models 3.3 and 4.1 often correlate in their accuracy. In general, the two power models performed at their best on the same server configuration and at their worst on the same one as well. This is possibly due to their nature of being odd polynomials. The power model 3.3 has an R-squared value greater than 0.95 in three out of eight configurations as well.

Our asymptotic power model 4.2 predicts the third most accurately when compared to the other power models in seven out of eight cases. It predicts more accurately than 3.2 in each case. In the configuration where model 3.3 of Zhang et al. [ZLQZ13] and our polynomial model 4.1 produce the already mentioned exceptionally inaccurate prediction, it is the most accurate power model. The R-squared values for 4.2 have a range from 0.521 to 0.974. The lowest RMSE value is 1.52 watts, the highest is 18.16. While individual results are not very accurate here, our asymptotic model 4.2 does not have a configuration in which its predictions lead to R-squared values in the negative or with an RMSE of above 20, like all other power models do. So it makes more stable predictions than the other power models while having only two configurations where it has an R-squared value of over 0.95.

The power model 3.2 of Fan et al. [FWB07] again performs least accurately in all but the exceptional configuration that was already mentioned. In this configuration, it ranks second in terms of accuracy. R-squared values range from -0.605 to 0.939. RMSE values range from 4.4 watts to 31.81 watts.

We need to point out that in all the server configurations with the exception of: BL460c Xeon E5-2640, SY480 Xeon Gold 6132, and DL380 Xeon E5-2690v4, we have to resort to using the maximum power as reported by the OneView API instead of averaged samples from the dataset. We do this because the servers that we consider here were never fully utilised. As we discuss in Section 7.5 on page 56, taking the maximum power from the OneView API might introduce a significant error to this power model's prediction. Nonetheless, the relative accuracy of 3.2 to the other power models remains the same in the three configurations, in which we did obtain power data with corresponding CPU utilisations of above 97 percent.

6.5.2 Training on server types combined based on their CPU model

Next, we combine the sever models presented in the last section based on their CPU model and retrain the models on these combinations. We do this to provide insight into how generalisable these power models are and which hardware characteristics of the servers can be disregarded when clustering them into configurations for power model training. We display these four configurations in Table 6.4.

Our polynomial power model

Our polynomial power model 4.1 predicts reasonably accurately in these configurations, with R-squared values ranging from 0.866 to 0.988 and RMSE values from 4.17 to 28.81 watts. This makes it the most accurate power model in each configuration again, purely judging by the two metrics. However, it makes one semantically invalid prediction again, whereas the other power

server-type	cpu-model	cpu-freq	core-count	storage	server-count	r-sq	rmse	power-model
DL380	X.E5-2620v3	2.4GHz	6	64GB	2	0.950	3.98	Asympt.
DL380	X.E5-2620v3	2.4GHz	6	64GB	2	0.939	4.4	Fan et al.
DL380	X.E5-2620v3	2.4GHz	6	64GB	2	0.978	2.65	Polynomial
DL380	X.E5-2620v3	2.4GHz	6	64GB	2	0.974	2.87	Zhang et al.
Ml350	X.E5-2620v3	2.4GHz	6	32GB	2	0.882	1.52	Asympt.
Ml350	X.E5-2620v3	2.4GHz	6	32GB	2	−0.605	5.6	Fan et al.
Ml350	X.E5-2620v3	2.40GHz	6	32GB	2	0.972	0.74	Polynomial
Ml350	X.E5-2620v3	2.4GHz	6	32GB	2	0.966	0.82	Zhang et al.
BL460c	X.E5-2640	2.50GHz	6	Without	3	0.870	13.25	Asympt.
BL460c	X.E5-2640	2.50GHz	6	Without	3	0.687	20.59	Fan et al.
BL460c	X.E5-2640	2.50GHz	6	Without	3	0.941	8.95	Polynomial
BL460c	X.E5-2640	2.50GHz	6	Without	3	0.887	12.34	Zhang et al.
DL360p	X.E5-2640	2.5GHz	6	32GB	1	0.521	4.94	Asympt.
DL360p	X.E5-2640	2.5GHz	6	32GB	1	0.247	6.2	Fan et al.
DL360p	X.E5-2640	2.5GHz	6	32GB	1	0.663	4.14	Polynomial
DL360p	X.E5-2640	2.5GHz	6	32GB	1	0.654	4.2	Zhang et al.
BL460c	X.E5-2690v4	2.6GHz	14	512GB	1	0.819	12.67	Asympt.
BL460c	X.E5-2690v4	2.6GHz	14	512GB	1	0.471	21.65	Fan et al.
BL460c	X.E5-2690v4	2.6GHz	14	512GB	1	−39.927	190.35	Polynomial
BL460c	X.E5-2690v4	2.6GHz	14	512GB	1	−0.014	29.97	Zhang et al.
DL380	X.E5-2690v4	2.6Ghz	14	384GB	1	0.665	18.16	Asympt.
DL380	X.E5-2690v4	2.6Ghz	14	384GB	1	0.354	25.21	Fan et al.
DL380	X.E5-2690v4	2.6Ghz	14	384GB	1	0.878	10.94	Polynomial
DL380	X.E5-2690v4	2.6Ghz	14	384GB	1	0.814	13.52	Zhang et al.
BL460c	X.G.6132	2.6GHz	14	1024GB	4	0.974	5.93	Asympt.
BL460c	X.G.6132	2.6GHz	14	1024GB	4	0.388	29.04	Fan et al.
BL460c	X.G.6132	2.6GHz	14	1024GB	4	0.996	2.41	Polynomial
BL460c	X.G.6132	2.6GHz	14	1024GB	4	0.981	5.15	Zhang et al.
SY480	X.G.6132	2.6GHz	14	Without	21	0.711	17.39	Asympt.
SY480	X.G.6132	2.6GHz	14	Without	21	0.032	31.81	Fan et al.
SY480	X.G.6132	2.6GHz	14	Without	21	0.945	7.61	Polynomial
SY480	X.G.6132	2.6GHz	14	Without	21	0.866	11.84	Zhang et al.

Table 6.3: Server models with same CPU, treated as different configurations.

models do not, as depicted in Figure 6.9. We will discuss the behaviour of the different power models when trained on sparse datasets more in Section 7.4. Our polynomial power model 4.1 has only one good fit with an R-squared value of above 0.95. It is the only power model to obtain such a high R-squared value in these configurations.

How did the accuracy of our polynomial power model 4.1 change in this combination, compared to its predictions on the individual server types in Table 6.3? We observe that in one of the four configurations, power model 4.1 predicts unambiguously less accurately than in the underlying two server types in Table 6.3. In the three other configurations of Table 6.4, our polynomial power model 4.1 is more accurate than it is in one of the underlying server types in Table 6.3 and less

cpu-model	gen	cpu-freq	core-count	server-count	r-sq	rmse	power-model
X.E5-2640	gen8	2.5GHz	6	4	0.928	10.49	Polynomial
X.E5-2640	gen8	2.5GHz	6	4	0.862	14.57	Zhang et al.
X.E5-2640	gen8	2.5GHz	6	4	0.642	23.48	Fan et al.
X.E5-2640	gen8	2.5GHz	6	4	0.836	15.87	Asympt.
X.E5-2620	gen9	2.5GHz	6	4	0.746	39.65	Asympt.
X.E5-2620	gen9	2.5GHz	6	4	0.822	33.23	Zhang et al.
X.E5-2620	gen9	2.5GHz	6	4	0.866	28.81	Polynomial
X.E5-2620	gen9	2.5GHz	6	4	0.074	75.76	Fan et al.
X.E5-2690v4	gen9	2.6GHz	14	2	0.811	22.15	Zhang et al.
X.E5-2690v4	gen9	2.6GHz	14	2	0.906	15.58	Polynomial
X.E5-2690v4	gen9	2.6GHz	14	2	0.615	31.58	Asympt.
X.E5-2690v4	gen9	2.6GHz	14	2	0.108	48.11	Fan et al.
X.G.6132	gen10	2.6GHz	14	25	0.924	10.43	Zhang et al.
X.G.6132	gen10	2.6GHz	14	25	0.988	4.17	Polynomial
X.G.6132	gen10	2.6GHz	14	25	0.837	15.27	Asympt.
X.G.6132	gen10	2.6GHz	14	25	0.357	30.34	Fan et al.

Table 6.4: server types with same CPU, treated as one configuration.

accurate than it is in the other. In these cases, we take the averages of both RMSE and R-squared for the two underlying server types in Table 6.3. We then compare this average to the metrics we obtained for the corresponding combined server configuration in Table 6.4.

In the case of the CPU model Xeon E5-2640, the average of the R-squared values of power model 4.1 in the two server types with this CPU in Table 6.3 is 0.802 and the averaged RMSE is 6.545, which compares to an R-squared value of 0.928 and an RMSE of 10.49 in the combined configuration in Table 6.4. So a larger portion of the variance in the dataset is explainable by the CPU utilisation to power consumption as expressed in power model 4.1, but the overall average distance of the prediction to the actual values increased slightly. So this result remains ambiguous.

In the configuration for CPU model Xeon E5-2690v4, we again take the average of the accuracy of power model 4.1 in the underlying two server types of Table 6.3. The averaged R-squared value is negative 39.108 because in one of the two server configurations power model 4.1 predicts very inaccurately. This also leads to an average RMSE value of 100.645 watts. Here, because the predictions of the power model 4.1 in one of the underlying two server configurations is so inaccurate, the accuracy of the same power model on the configuration that combines these server types is more accurate. Power model 4.1 has an R-squared value of 0.906 here and an RMSE value of 15.58 watts, so in Table 6.4, the accuracy of the polynomial model 4.1 improved.

In the configuration with CPU model Xeon Gold 6132, the averaged metrics for power model 4.1 are 0.9705 as the R-squared value and 5.01 watts as the RMSE. This compares to 0.988 and 4.17 in the combined configuration in Table 6.4. So here the accuracy also went up, when compared to the average of the accuracy metrics that power model 4.1 has on the two individually considered server types.

Summing up, when compared to the average of the accuracy metrics our polynomial power model 4.1 has in the server configurations in Table 6.3, the accuracy of our polynomial power model 4.1 improves in two cases, diminished in one case and sees mixed results in the remaining case.

Power model 3.3 of Zhang et al. [ZLQZ13]

The power model 3.3 of Zhang et al. [ZLQZ13] predicts the second most accurately in each configuration again, judging by the metrics. The R-squared values for this power model range from 0.811 to 0.924 and the RMSE values are in the range from 10.43 to 33.23 watts. It does not predict power consumption to be negative in any of the four configurations.

When comparing the accuracy of the power model 3.3 in Table 6.4 to the accuracy it has in the corresponding server types in Table 6.3, we observe that the accuracy diminished in two cases, while having results that need to be further analysed in the remaining two cases.

In one of these two cases, namely the CPU model Xeon E5-2640, the averaged metric values that power model 3.3 of Zhang et al. [ZLQZ13] has on the corresponding server types in Table 6.3 are 0.7705 as the R-squared value and 8.27 as the RMSE value. We compare this to the R-squared value of 0.862 and the RMSE value of 14.57 that power model 3.3 has in the corresponding combined configuration in Table 6.4. This result remains ambiguous.

In the other case, where the CPU model Xeon Gold 6132 is concerned, the averaged metrics are 0.9235 as R-squared and 8.495 as the RMSE value. In Table 6.4, power model 3.3 of Zhang et al. [ZLQZ13] has an R-squared value of 0.924 and an RMSE value of 10.43 on the corresponding configuration. So the power model 3.3 predicts unambiguously less accurately in the combined configuration in Table 6.4, than it did in the average of the two corresponding configurations in Table 6.3.

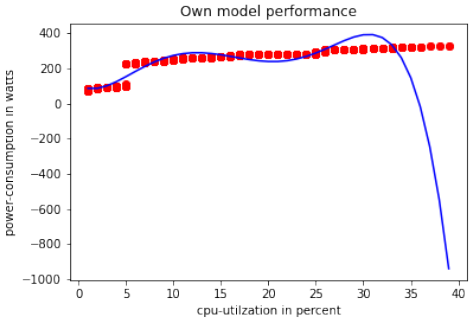
In conclusion, the power model 3.3 of Zhang et al. [ZLQZ13] predicts less accurately when server types are combined in three cases and sees mixed results in the remaining case.

Asymptotic power model

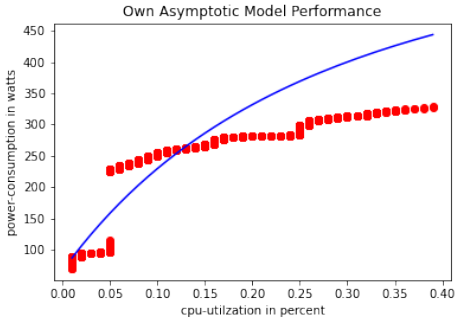
Our asymptotic power model 4.2 has relatively low accuracy in most of the four configurations, with the R-squared values being in the range from 0.615 to 0.837. The RMSE values are in the range of 15.27 to 39.65 watts. This makes the asymptotic power model 4.2 the third-most accurate model in these configurations, based on a comparison of the two metrics. However, the power model 4.2 does not predict semantically incorrect values in any of the configurations. That is to say, no negative values are predicted by 4.2 and the power is also not predicted to decrease with increasing CPU utilisation.

When comparing the accuracy that our asymptotic power model 4.2 has here to its accuracy in Table 6.3, we observe that in two of the four configurations, power model 4.2 predicts unambiguously less accurately than in the underlying two server types in Table 6.3.

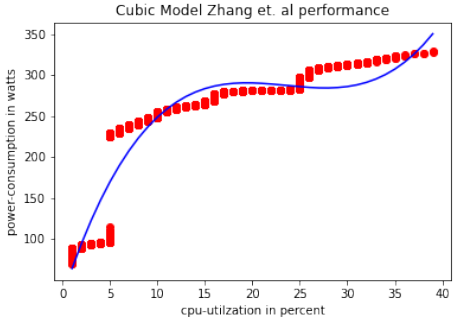
In the two remaining configurations of Table 6.4, the already mentioned results appear, where power model 4.2 is more accurate here than on one of the underlying server types but less accurate than it is on the other.



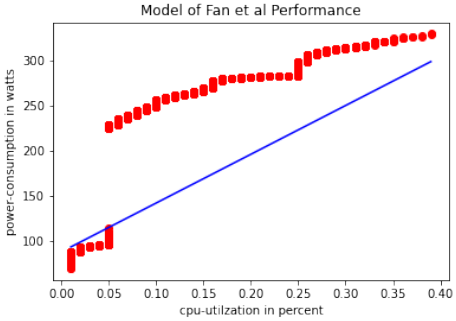
(a) Our Polynomial power model 4.1



(b) Our Asymptotic power model 4.2



(c) Power model 3.3 of Zhang et al. [ZLQZ13] equation: 3.3.



(d) Power model of Fan et al. [FWB07] equation 3.2.

Figure 6.9: Semantically incorrect prediction of Power model 4.1 on combined dataset of CPU model Xeon E5-2620.

In the case of the CPU model Xeon E5-2640, the average of the R-squared values of the two server types with this CPU in Table 6.3 is 0.6955 and the averaged RMSE is 12.96, which compares to an R-squared value of 0.836 and an RMSE of 15.87 in the combined configuration in Table 6.4. So a larger portion of the variance in the dataset is explainable by the CPU utilisation to power consumption as expressed in power model 4.2, but the overall average distance of the prediction to the actual values increases slightly. This result remains ambiguous.

In the two servers with CPU Xeon Gold 6132, we average the two metrics for our asymptotic power model 4.2 as well. The averaged R-squared value is 0.8425, while the averaged RMSE is 11.66. When comparing this to the corresponding combined server configuration in Table 6.4, we observe that the R-squared value is 0.837 and the RMSE is 15.27. So both metrics indicate a less accurate prediction by the power model 4.2 in the combined configuration.

In three out of four configurations the accuracy of power model 4.2 diminished and sees a mixed result in the remaining configuration when compared to the corresponding results out of Table 6.3.

Power model of Fan et al. [FWB07]

The power model 3.2 of Fan et al. [FWB07] has R-squared values in the range from 0.074 to 0.642 and RMSE values in the range of 23.48 to 75.76 watts. This makes it the least accurate model again. However, like power models 4.2 and 3.3, it does not predict semantically incorrect values in any of the four configurations.

When comparing the accuracy of prediction of the power model 3.2 of Fan et al. [FWB07] in Table 6.4 to those in Table 6.3, we observe that it predicts unambiguously less accurately in one instance, while the other three configurations need to be analysed further.

When considering the CPU model Xeon E5-2640, the averaged metrics of model 3.2 out of Table 6.3 are 0.467 as R-squared and 13.395 as RMSE value. We compare this to the metric values of 0.642 as R-squared and 23.48 that power model 3.2 of Fan et al. [FWB07] has on the corresponding combined configuration in Table 6.4. This comparison does not allow for a clear assessment, in which case the power model 3.2 of Fan et al. [FWB07] predicts more accurately.

In the case of the CPU model Xeon E5-2620, the averaged metrics are 0.167 as the R-squared value and five watts as the RMSE. This indicates a more accurate prediction than power model 3.2 of Fan et al. [FWB07] makes in the corresponding combined configuration in Table 6.4, in which it has an R-squared value of 0.074 and an RMSE value of 75.76 watts.

The third CPU model that - at first glance- leads to ambiguous result for power model 3.2 of Fan et al. [FWB07] is Xeon Gold 6132. Here, the averaged metric values are 0.21 as the R-squared value and 30.425 watts as the RMSE value. This indicates that power model 3.2 of Fan et al. [FWB07] makes a more accurate prediction in the corresponding combined configuration in Table 6.4 because the R-squared value is 0.357 here and the RMSE value is 30.34 watts.

In conclusion, out of the four configurations in which we combine server types based on their CPU model, power model 3.2 of Fan et al. [FWB07] makes less accurate predictions in one case, on average more accurate predictions in two other cases, while the remaining case remains ambiguous.

6.6 Training on the full dataset

To further test the power models on their generalisability, in this section, train and test the power models on the entire dataset of 73 servers. So no categories of servers are made here whatseover. All differences within the servers like server type, CPU model, storage size, core count, and so on are disregarded.

If the power models predict accurate results here, it would point towards greater generalisability and to the potential of being able to apply the power models on different hardware, particularly when they were trained on large datasets, without requiring retraining.

We display the accuracy metrics of the four power models on the entire dataset in Table 6.5 and the corresponding graphical representations of the predictions in Figure 6.10.

We observe that our polynomial power model 4.1 is the most accurate in its predictions once again with an R-squared value of 0.971 and an average error of only 16.89 watts. These metrics are confirmed when analysing Figure 6.10b. This time power model 4.1 does not predict semantically incorrect values.

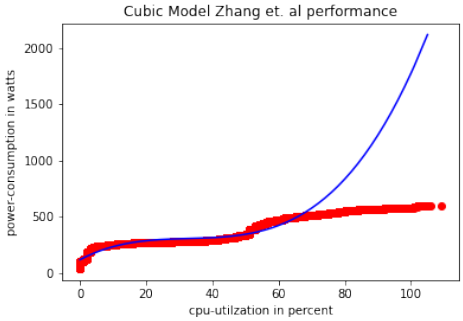
The next most accurate power model is again 3.3 of Zhang et al. [ZLQZ13]. It has an R-squared value of 0.871 and an RMSE value of 35.95 watts. In Figure 6.10a, we see that the power model 3.3 drastically overpredicts the power consumption in the range from 80-100 percent utilisation with the maximum difference being almost 1500 watts. In the lower CPU utilisations, it is almost perfectly accurate though. As we recall, the dataset is much denser in lower CPU utilisation ranges, as visible in Figure 6.1. This might be the reason why the high discrepancy between predicted values and actual measurements in the higher utilisation ranges has not such a big influence on the overall accuracy of the power model 3.3 by Zhang et al. [ZLQZ13].

Our asymptotic power model 4.2 seems to be having a smaller error at first glance in the graph in Figure 6.10d but comes in as the third most accurate power model again in Table 6.5. This is probably due to power model 4.2 not following the actual power curve as well as power models 4.1 and 3.3 do in the lower percentage utilisations.

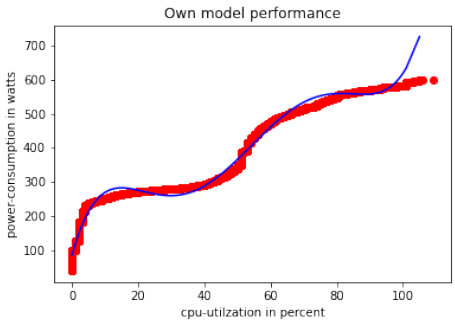
This is also true for the power model 3.2 of Fan et al. [FWB07], which seems more accurate than ou asymptotic model 4.2 and model 3.3 of Zhang et al. [ZLQZ13] when we consider the graph in Subfigure 6.10c. The higher utilisation ranges from 60 to 100 percent utilisation are followed very well by power model 3.2 of Fan et al. [FWB07]. However, in the lower percentage utilisation ranges, the power model 3.2 by Fan et al. [FWB07] has a larger discrepancy than the other models and due to the already mentioned makeup of the dataset, as displayed in 6.1, this leads to the undesirable metric results for 3.2.

server-count	r-sq	rmse	power-model
73	0.971	16.89	Polynomial
73	0.871	35.95	Zhang et al.
73	0.785	46.32	Asympt.
73	0.609	62.5	Fan et al.

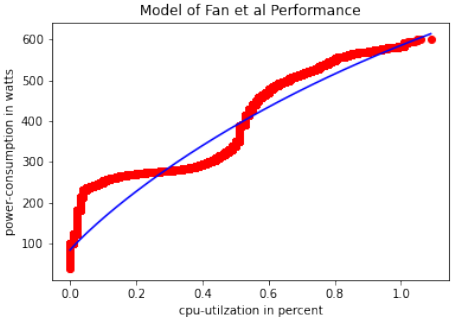
Table 6.5: Power models trained on the entire dataset, making not distinctions whatsoever.



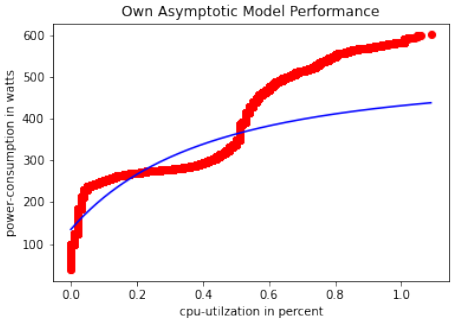
(a) Power model 3.3 of Zhang et al. [ZLQZ13] equation 3.3.



(b) Our Polynomial power model 4.1.



(c) Power model of Fan et al. [FWB07] equation 3.2



(d) Asymptotic Model 4.2.

Figure 6.10: Predictions of the power models on the entire dataset of 73 servers.

7 Discussion of Results

In the last chapter, we displayed the results of our evaluation. In this chapter, we want to discuss how accurately the power models predict in comparison to each other, to what extent they are generalisable, and how fast they can be trained.

Thereafter in Section 7.4, we discuss the way the models behave when the dataset is very sparse, which is very often the case in our analysis. Subsequently, we discuss what influence our usage of the *calibrated maximum power* value from the OneView API in training the power model 3.2 of Fan et al. [FWB07] and our asymptotic model 4.2 might have on their accuracy in Section 7.5. And finally, in Section 7.6 we discuss the inadequacy of the R-squared metric when it is used to compare the fit of models which are non-linear in their parameters.

7.1 Power model accuracy

In the previous chapter we showed our evaluation of the four server power models 3.2, 3.3, 4.2 and 4.1. We now want to discuss their accuracy in some more depth. As a first observation, the accuracy the four power models have relative to each other remains surprisingly consistent throughout our analysis. Generally our polynomial model 4.1 predicts most accurately, power model 3.3 of Zhang et al. [ZLQZ13] predicts second most accurately, our asymptotic model 4.2 third most accurately and power model 3.2 of Fan et al. [FWB07] predicts least accurately. However, our polynomial model 4.1 and model 3.3 of Zhang et al. [ZLQZ13] sometimes predict nonsensical results in high utilisation ranges, which we will analyse further in Section 7.4.

7.1.1 Accuracy of our Polynomial Model 4.1

The polynomial model 4.1 we propose here predicts most accurately when considering the accuracy metrics in almost every case but occasionally predicts nonsensical results in higher CPU utilisation ranges that we will discuss further in Section 7.4. This accuracy relative to the other power models is generally true for the configurations we analyse in our evaluation but also for the ones we include in our Appendix A.

However, it is important to note that, while our polynomial power model 4.1 is ,in general, the most accurate model in our analysis when compared to the other three power models, it does have many configurations, on which it does not have an R-squared value of greater than 0.95. We take this value to be an indicator for when a model is a "good fitänd our polynomial power model 4.1 is not a good fit in many of the configurations.

This could be influenced by the sparseness of the dataset in higher utilisation ranges as seen in the histogram of the dataset 6.1, but we cannot be sure of this.

So while the relative accuracy of this model is remarkable, when we compare it to the other three, the predictions of this model need to be checked for their semantic correctness and in many configurations, it is not a good fit for the dataset.

7.1.2 Accuracy of model 3.3 of Zhang et al. [ZLQZ13]

As already mentioned, the power model 3.3 of Zhang et al. [ZLQZ13] is generally the second most accurate model. In some configurations, it does predict a sharp increase or decrease in power consumption when approaching full utilisation. This makes the predictions of this power model less accurate in this range. But this result may be different if the underlying utilisation dataset were to be more evenly distributed.

7.1.3 Accuracy of our asymptotic model 4.2

Our asymptotic model 4.2 does not predict as accurately as the two other models already mentioned here but does not have the same semantic problems. It generally follows the trend of the power curves well, predicting no sharp increase or decrease of power consumption at full utilisation, which is the trend we usually observe in the plots of our server configurations.

But due to its frequent over- or underprediction, this model is less accurate in general. Our asymptotic model also depends on power consumption measurements at full utilisation as an initial guess, which might influence its accuracy negatively. We will discuss this further in Section 7.5.

7.1.4 Accuracy of model 3.2 of Fan et al. [FWB07]

In general, the power model 3.2 of Fan et al. [FWB07] predicts least accurately. This is true, both when RMSE and R-squared are considered. It has only a few configurations where it predicts with an R-squared value of above 0.95. However, as already mentioned, we have to resort in most configurations to take the maximum power consumption as reported by the OneView API as the maximum power value that power model 3.2 of Fan et al. [FWB07] requires. This is not how the authors of [FWB07] trained their model and it might be the source of an error, which we will discuss more in Section 7.5.

7.2 Power model generalisability

In our evaluation 6, we train the power models on different subsets of the entire dataset, to find out which hardware characteristics are important to consider when categorising servers into groups for training.

We train the power models with storage size consideration and without, to find out if this difference can be disregarded within the same server type. Likewise, we train the power models on a combination of server types that have the same CPU model and on each of those server types individually. Finally, we train the models on the entire dataset, disregarding all hardware characteristics.

In this section, we want to discuss how the power model accuracy develops when some or all hardware characteristics are disregarded. The less fine-grained the distinctions between configurations need to be, for the power models to predict accurate results, the better.

If all hardware characteristics of the servers need to be considered for the power models to predict accurately, then each new server that is deployed with a slight difference in hardware characteristics requires new data gathering and power model retraining. Also, the power model predictions probably need to be aggregated in some way, maybe for optimisation of scheduling algorithms, or in the form of a dashboard. This becomes increasingly difficult, when the number of trained power models rises.

So if the power models were to be generalisable in this sense, it would make their application in data centers where there is a lot of diversity in the hardware deployed, for example the one in AEB, much less labour intensive. The broader the categories for forming configurations can be while maintaining an acceptable accuracy in power model prediction, the better.

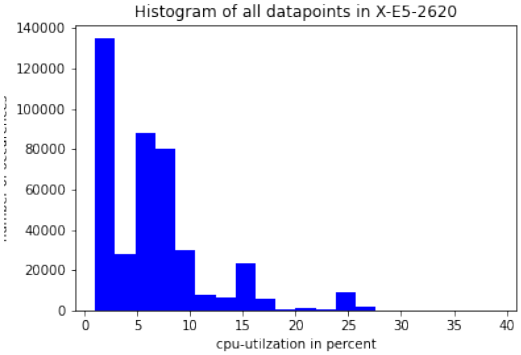
We will discuss the possibility of disregarding each hardware characteristic in the order that they appear in our evaluation, starting with storage size.

7.2.1 Can storage size be disregarded?

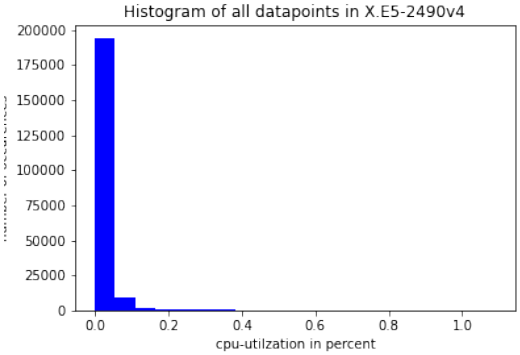
In all power models, the amount of configurations that result in improved accuracy when storage is considered is significantly larger than the amount of less accurate and ambiguous results combined. So it is reasonable to say that it generally improves the accuracy of the four power models when storage is considered in categorising servers into configurations for power model training. This is especially true, if the resulting configurations are not too sparse, as the configurations where only three or fewer servers of this kind are left in the dataset, are more often subject to worsened or ambiguous results than those with more servers in the dataset.

While the models deliver more accurate results in most cases where storage size is considered, they still deliver useful predictions in some cases where these differences are disregarded. Especially our polynomial power model 4.1 delivers accurate results in the configurations of Table 6.1, with all its R-squared values being equal to or greater than 0.941 and with three configurations with a value greater than 0.95. Power model 3.3 delivers quite accurate results as well, with only two R-squared values beneath 0.9 and one configuration with an R-squared value greater than 0.95. Power models 4.2 and 3.2 each do have one good fit with an R-squared value of above 0.95 but are in general less accurate than the other two models.

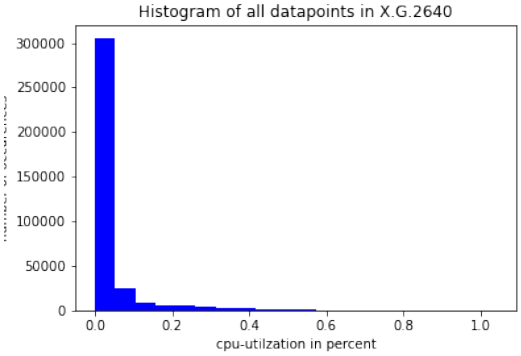
These results point towards especially our polynomial power model 4.1 being useful even if it is not trained individually on each server model and storage size combination but rather just on server models, disregarding storage differences. We argue that 3.3 also shows promise for being generalisable in this sense, but power models 4.2 and 3.2 of Fan et al. [FWB07] seem to require storage consideration to deliver more accurate results.



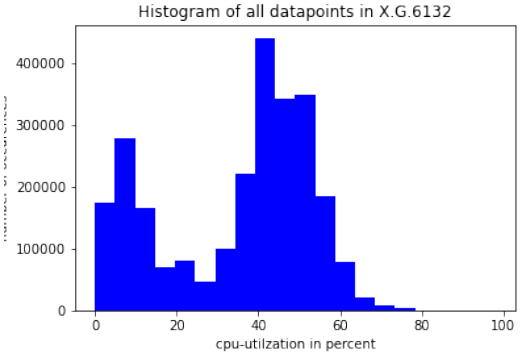
(a) Configuration with CPU model X.E5-2620.



(b) Configuration with CPU model X.E5-2490v4.



(c) Configuration with CPU model X.E5-2640.



(d) Configuration with CPU model X.G.6132.

Figure 7.1: Histograms of the configurations in Table 6.4.

7.2.2 Can server types be combined based on their CPU?

When the server types are combined into one dataset based on their CPU model, the accuracy of the model 3.3 of Zhang et al. [ZLQZ13] and that of our asymptotic power model 4.2 diminish. The results for the remaining two power models 4.1 and 3.2 are mixed, with in each case two configurations where the accuracy went up, one where it went down, and one where the results remained ambiguous.

I conclusion: if the power models can be trained on sufficiently large datasets for individual server types, they should be trained separately. For power models 3.2 and 4.1 it might be a possibility to combine the server types based on their CPU model, but this needs to be evaluated in each individual case.

However, it is good to note that the datasets are heavily biased towards low CPU utilisation. In a dataset that has more data points across all CPU utilisation ranges, the power models might predict more accurate results. This bias in the datasets is displayed in Figure 7.1, where the CPU utilisation histograms of the configurations in Table 6.4 on page 43 are visible.

7.2.3 Can all hardware characteristics be disregarded?

In Section 6.6 on page 46, we train the power models on the entire dataset of over seven million data points. Although the results of the previous two sections do not indicate that hardware characteristics can be disregarded, this might be due to the relatively small datasets, which are also heavily biased towards lower utilisation ranges.

While the results of training the power models on the entire dataset are not perfect, they are much better than we anticipated and point towards high generalisability of especially our polynomial power model 4.1. When it is trained on large datasets, its predictions seem to be useful for a great variety of server types. If the dataset were to be more evenly distributed, judging by the graph in Figure 6.10 on page 48, power model 3.2 of Fan et al. [FWB07] would have made quite useful predictions as well here. Our asymptotic power model 4.2 and the power model 3.3 of Zhang et al. [ZLQZ13] both do not generalise well to the entire dataset, with the asymptotic model underpredicting and the model of Zhang et al. [ZLQZ13] overpredicting the consumed power in higher CPU utilisation ranges.

7.3 Speed of model training

Two of the models, namely our polynomial power model 4.1 and model 3.3 of Zhang et al. [ZLQZ13] can be trained quite quickly, even when using cautious k-fold cross-validation, as we do. The k-fold cross-validation step takes only fifteen to twenty minutes when training the models on the entire dataset of 7,308,761 non-null values.

On the same dataset, our asymptotic model 4.2 and 3.2 of Fan et al. [FWB07] both take over two hours to be trained. We train these power models with the curve_fit [Scid] function of the python library scikit learn and apparently this function has a much higher runtime than the simple linear regression function called fit [Scif] that we use in the training of our polynomial model 4.1 and 3.3 of Zhang et al. [ZLQZ13].

7.4 Performance in the presence of sparse data

When judged by the metrics, generally the two power models 4.1 and 3.3 are the two most accurate, but when analysing the graphical representation of their predictions, we sometimes notice semantic problems. Models 3.3 and especially 4.1 predict a negative power consumption for higher CPU utilisations in some server configurations.

While in general, these semantically incorrect predictions do not prevent these two power models from being the most accurate models purely judging from the metrics, in one case they do. In Table 6.3 we find the configuration BL460c with Cpu Xeon Ej5-2690 v4, that sees negative R-squared values for both 4.1 and 3.3, as well as high RMSE values, with our polynomial power model 4.1 having an error of 190.35 watts.

We will now take a look at this exceptional configuration, where power model 4.1 predicts least accurately, while our asymptotic model 4.2 predicts most accurately of all four power models. The predictions of all four models as well as the actual power data at a given CPU utilisation are depicted

in Figure 7.2. The underlying dataset is again very sparse, with the majority of the data points being in the range from zero to twenty percent CPU utilisation, followed by a gap ranging from twenty-three percent to 40 percent utilisation. No data above 45 percent CPU utilisation is available as depicted in the histogram of this configuration in Figure 7.3.

Both our polynomial power model 4.1 and model 3.3 of Zhang et al. [ZLQZ13] predict a negative power consumption. This is a semantic problem, as servers are not a power source. And it also leads to a high error because the small part of the dataset which is at a CPU utilisation of around 40 percent has a large distance to the predicted values.

In the case of our polynomial model 4.1 the predicted graph would be at a value of a few negative thousand watts at that point, should the trajectory of the graph continue after the point where the Figure 7.2 cuts off. In the case of 3.3 the prediction of power consumption would be in the scope of negative 1000 watts at 40 percent utilisation, again assuming the trajectory remains the same. These results are significant outliers that influence the result a lot when an average is taken. Thus the exceptional metric results can be explained for models 4.1 and 3.3 of Zhang et al. [ZLQZ13] .

The power model 3.2 of Fan et al. [FWB07] predicts a linearly increasing power consumption here, with the highest predicted value being around 360 watts and the lowest predicted value being 175 watts. The prediction of the power model 3.2 is very close to the actual values in the range from three to fifteen percent. Before that the power at idle CPU is over-estimated by power model 3.2 of Fan et al. [FWB07] by 50-75 watts and the highest value at 40 percent CPU utilisation over-estimates the actual value by approximately 50 watts. While the prediction may not be very accurate in some portions of the dataset, it is semantically valid.

It is not immediately obvious that our asymptotic power model 4.2 is more accurate than model 3.2 of Fan et al. [FWB07] in this case, as the asymptotic model 4.2 starts to over-estimate the actual consumed power at around five percent utilisation and keeps on doing so for the remainder of the dataset. However, when considering the histogram of CPU utilisation in this configuration as depicted in Figure 7.3, the reason becomes clearer.

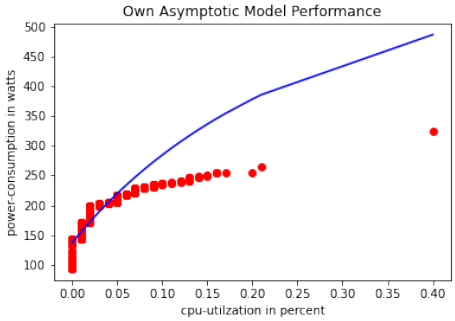
Over 90 percent of the total dataset is in the range of zero to five percent CPU utilisation. Here, judging by the graphs in Figure 7.2, asymptotic power model 4.2 predicts the power consumption quite accurately, while model 3.2 of Fan et al. [FWB07] over-estimates the consumed power in this range. Like power model 3.2 of Fan et al. [FWB07], our asymptotic power model 4.2 predicts semantically correct values in this configuration.

Both our polynomial power model and the model 3.3 of Zhang et al. [ZLQZ13] are prone to predicting sharp increases in power consumption, when approaching full CPU utilisation and, in two instances, also a sharp decrease. Both datasets, where the semantic problem of predicted sharply decreasing power consumption occurs are very sparse as seen in 7.3 and 7.4

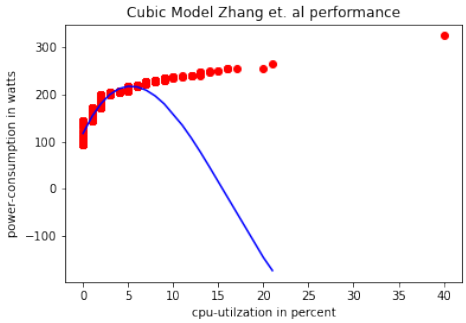
We display examples of these semantic errors in prediction of especially our polynomial power model 4.1 in Figure 7.5, where our polynomial model predicts a sharply decreasing power consumption, while power model 3.3 of Zhang et al. [ZLQZ13] predicts a sharp increase. Also note Figure 7.6, where both models predict a sharp increase, with our polynomial model also briefly predicting a decrease in power consumption.

It is important to note, that while the graphs look inaccurate, due to the bias in the datasets towards low percentage CPU utilisation, models 4.1 and 3.3 are generally still more accurate than the other two models in these cases, judging by the two metrics R-squared and RMSE. This is explainable

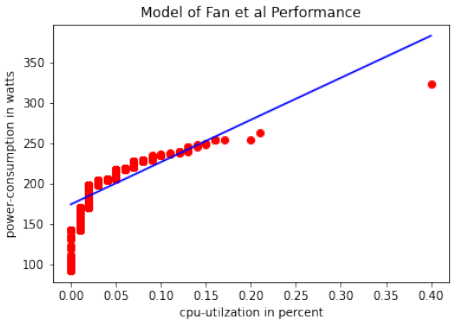
(a) Our Polynomial power model.



(b) Our Asymptotic power model.



(c) Power model 3.3 of Zhang et al. [ZLQZ13] equation: 3.3.



(d) Power model of Fan et al. [FWB07] equation 3.2.

Figure 7.2: Model predictions trained on BL460c with CPU model X.E5-2690.

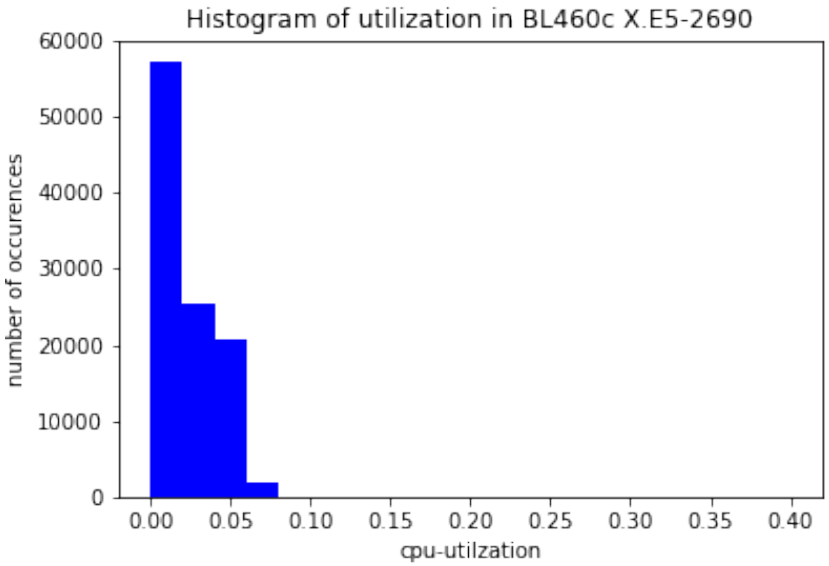


Figure 7.3: Histogram of the CPU utilisation in BL460c X.E5-2690v4.

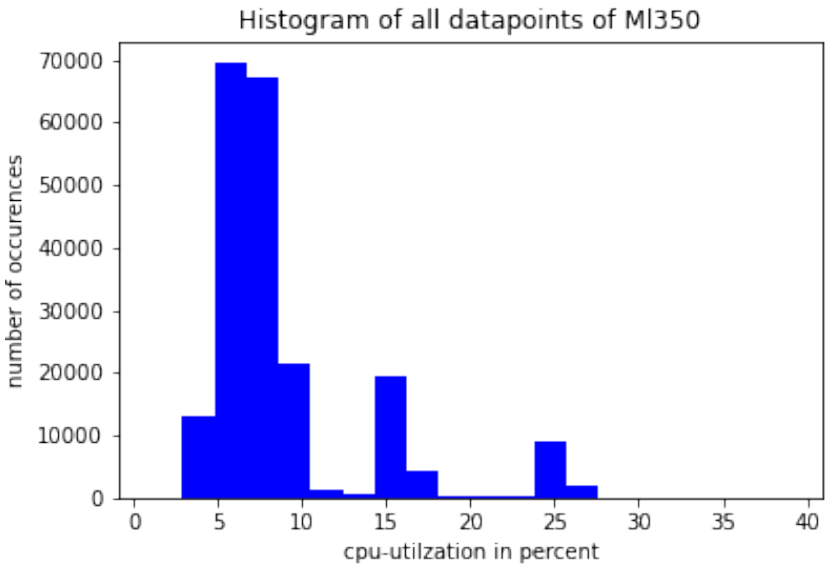


Figure 7.4: Histogram of the CPU utilisation in Ml350 with CPU model X. E5-2620v3.

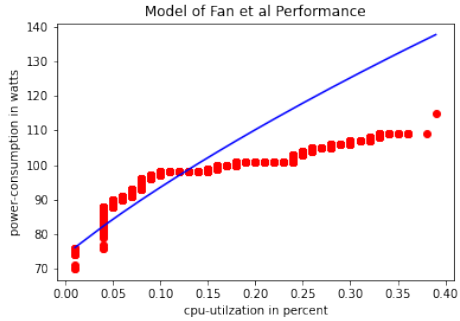
because their inaccuracies in the high percentage utilisation ranges have only a small impact. If the power models are accurate in low percentage utilisation ranges and the dataset is biased towards that range, then this has the bigger impact on the average.

We also want to stress here that, while nonsensical predictions do occur, they generally do so on very sparse datasets and they are the exception, not the rule. To showcase that all models have configurations, where they are a "good fit", we include Figure 7.7, where the models are trained on a configuration with a non-sparse dataset. All models predict with R-squared values greater than 0.95 (as seen in Table 6.2) and no nonsensical values appear. In our Appendix B, we also include the graphs of configurations we cannot discuss here, which include several such good fits.

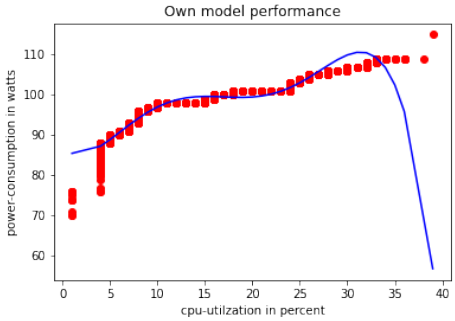
Nonetheless, these inaccuracies in higher utilisation areas are a problem for power models 3.3 and especially for our polynomial model 4.1. This needs to be addressed by visually checking the predictions that these models make because trusting the metrics is not enough here. The fact that our polynomial power model 4.1 predicts sharply decreasing power consumption on more configurations than 3.3 might be an indication of overfitting caused by the higher degree of 4.1.

7.5 Training with *calibrated maximum power* consumption

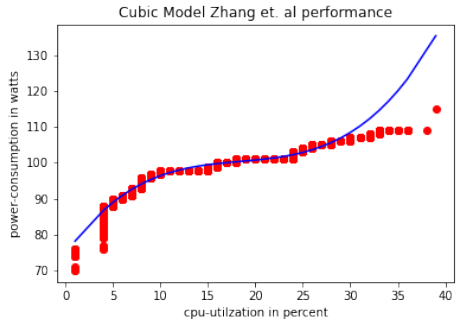
Fan et al. [FWB07] observe in their study that the maximum power consumption the manufacturers of the servers specify is significantly higher than the power actually measured at maximum CPU utilisation. Therefore, Fan et al. [FWB07] instead take the power observed at maximum CPU utilisation as the value P_{busy} in their power model. We follow this approach, when we have data points close enough to 100 percent CPU utilisation, and otherwise take the *calibrated maximum*



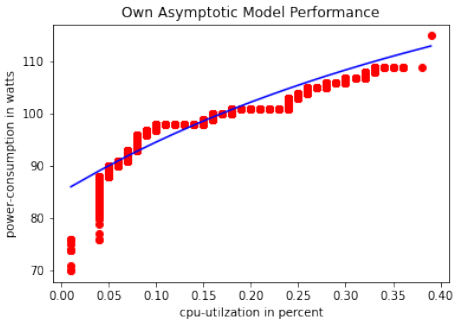
(a) Power model 3.2 of Fan et al. [FWB07].



(b) Our Polynomial power model 4.1.



(c) Power model 3.3 of Zhang et al. [ZLQZ13].



(d) Our Asymptotic power model 4.2.

Figure 7.5: On the sparse dataset of the server Ml350 with CPU model X.E5-2620v3, our polynomial power model 4.1 makes semantically incorrect predictions.

power consumption that the OneView API reports [Onea] as this value. This is not the same as the "nameplate power" Fan et al. [FWB07] talk about. The value of *calibrated maximum power* that the OneView API provides is calculated as follows:

"The calibrated maximum power.

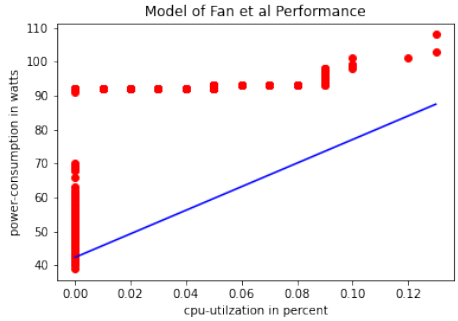
Calibrated Maximum Power is defined as the maximum potential power that the device can consume, subject to the following requirements and constraints:

1. The value reported MUST be the maximum which can be sustained for greater than 1/2 second (i.e., in-rush currents and other spikes that may persist for less than a 1/2 second are not to be included).

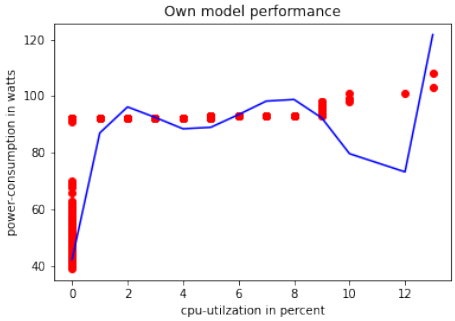
2. The value reported MUST represent the maximum total AC input across all power supplies

3. The value reported MUST represent the maximum AC input the device can sustain as configured at the time this metric is reported. If additional components are added later or if it is discovered at a later time that more power can be used, the larger number MUST be reported when the device is next queried for this metric.

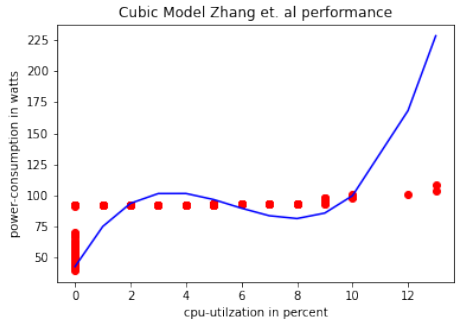
4. The value reported does not represent potential input power in the case of error conditions such as short circuits.



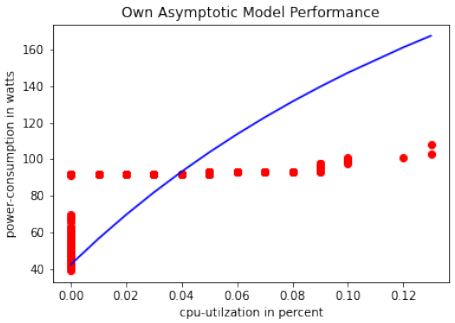
(a) Power model 3.2 of Fan et al. [FWB07].



(b) Our Polynomial power model 4.1.



(c) Power model 3.3 of Zhang et al. [ZLQZ13].



(d) Our Asymptotic power model 4.2.

Figure 7.6: On the sparse dataset of the server type DL360p with CPU model X.E5-2640, power models 4.1 and 3.3 make semantically incorrect predictions.

5. The actual power used by the device MUST NOT exceed the reported Calibrated Maximum Power by greater than 1%.

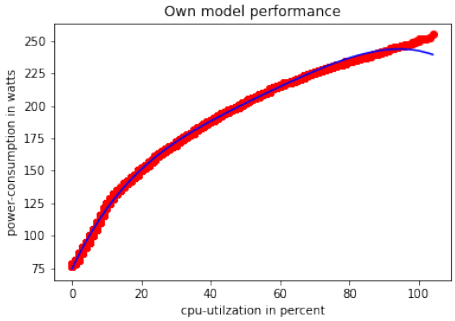
6. The Calibrated Maximum Power SHOULD NOT exceed the actual maximum power that the device is capable of using by more than 5 %.

This value will be calculated automatically for managed/monitored device." [Onea]

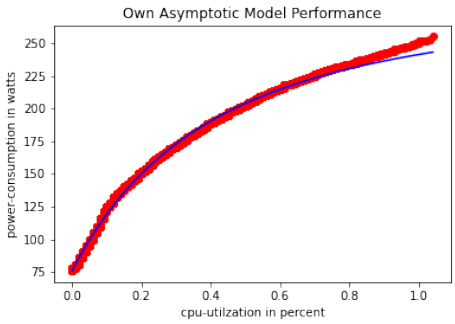
With servers yielding data close to 100 percent CPU utilisation, we observe a quite wide discrepancy between the *calibrated maximum power* value and the actual, averaged maximum power consumption of our measurements. We display this observation in Table 7.1. In the most striking case, the difference between the *calibrated maximum power* value and the actual average of the highest measured values is 826.25 watts. On average it overpredicts the actual value by 234.32 watts.

Apart from one outlier, our asymptotic power model makes predictions with an R-squared value of above 0.924 in all cases here. The power model 3.2 of Fan et al. [FWB07] also predicts quite accurately here, with three out of six configurations leading to a useful prediction with an R-squared value of over 0.95. Because of two outliers, the average R-squared is still quite low though, with a value of 0.694.

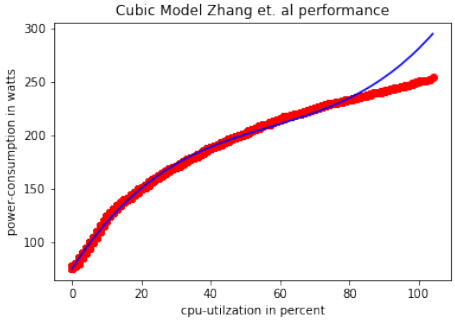
These six configurations contain some of the most accurate predictions of these two models, so it is fair to say, that the general lack in accuracy we observe in the case of the model 3.2 of Fan et al. [FWB07] is, at least in part, due to the bias we have in our dataset and the resulting use of the



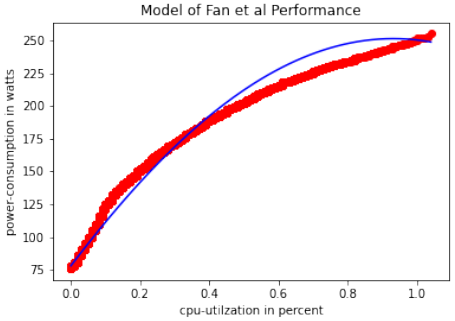
(a) Our Polynomial power model without storage consideration.



(b) Our Asymptotic power-mode.l



(c) Power model 3.3 of Zhang et al. [ZLQZ13] equation: 3.3.



(d) Power model of Fan et al. [FWB07] equation 3.2.

Figure 7.7: A less sparse dataset in higher utilisation area. Server configuration: BL460c X.E5-2640 with 128 gigabyte of storage.

calibrated maximum power value instead of real measurements. However, the two very inaccurate results this model produces even here, suggest that this might not be the only cause of this lack in accuracy that we observe.

There was one server type where the *calibrated maximum power* value was even further off. It comes in at 6000 watts for the model XL450, which we include in our Appendix A. If we recall the histogram of all power measurements 6.2, this means a ten-fold overestimation of the highest power consumption observed by us. We include the plots on this configuration in Figure 7.8 and want to especially highlight the trend towards a very high power consumption that the model 3.2 of Fan et al. [FWB07] predicts in the Subfigure 7.8d. Please note, that this is not one of the server types of Table 7.1, or a combination of server types where one of these is included, so we do use this high *calibrated maximum power* value to train power model 3.2 of Fan et al. [FWB07] and our asymptotic model.

It seems that this high value has a bigger impact on the prediction of the power model 3.2 of Fan et al. [FWB07] than it has on our asymptotic model. This is probably because this value is only an initial guess for the training of our asymptotic model 4.2 and gets readjusted during training. In the training of power model 3.2 of Fan et al. [FWB07] however, the power at maximum utilisation is taken to be a constant that is not further refined.

Some of the configurations of Table 7.1 are contained in the combined configurations when storage size is disregarded in Table 6.1 on page 31. These are the server types SY480 with CPU model Xeon Gold 6132, BL460c with CPU model Xeon E5-2640, and BL460c with CPU model Xeon E5-2660v3. The evaluation on these particular configurations is thus more valid than the remaining ones, where we resort to using *calibrated maximum power*.

Likewise, server types BL460c with CPU model Xeon E5-2640, SY480 with CPU model Xeon Gold 6132, and DL380 with CPU model Xeon E5-2690 version four are present in the table where they are combined based on their CPU model in Table 6.4 on page 43. Our evaluation on these configurations might likewise be more valid as well than on the remaining configurations, where we have to use the *calibrated maximum power* value of the OneView API.

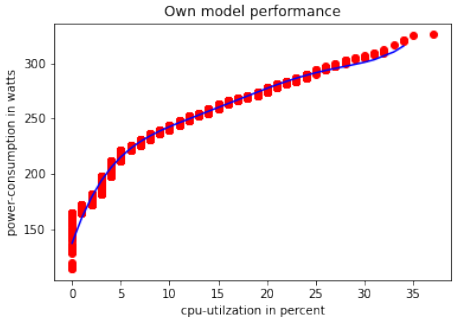
server-type	cpu-model	storage	measured-power	OneView-power	metrics-asymptotic	metrics-Fan-et-al
DL380	X.E5-2690v4	384GB	360.89	627	[0.665|18.16]	[0.354|25.21]
DL380	X.E5-2630v3	384GB	365.96	561	[0.963|7.48]	[0.902|12.12]
BL460c	X.E5-2660v3	1024GB	380.52	363.66	[0.961|4.686]	[0.011|23.45]
BL460C	X.E5-2640	128GB	250.6	291	[0.997|2.33]	[0.968|7.37]
SY480	X.G.6132	1024GB	444	471	[0.925|7.48]	[0.954|5.82]
DL360	X.G.6128	64GB	173.75	1,000	[0.974|1.80]	[0.977|1.69]

Table 7.1: The discrepancy of measured power versus the power the One View API reports. Metrics are [R-squared|RMSE]

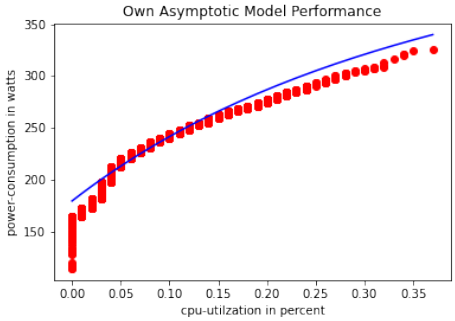
7.6 Suitability of R^2 for non-linear parameters

While evaluating the power models discussed, we notice that scipy, the library we use to train the power models, which are non-linear in their parameters, namely the one of Fan et al. [FWB07] 3.2 and our asymptotic model 4.2, does not provide an R-squared metric. Upon searching how to obtain this metric, we discover that the scipy community left this feature out intentionally [Int] because it is a poor basis for model comparison in the case of models that are non-linear in their parameters.

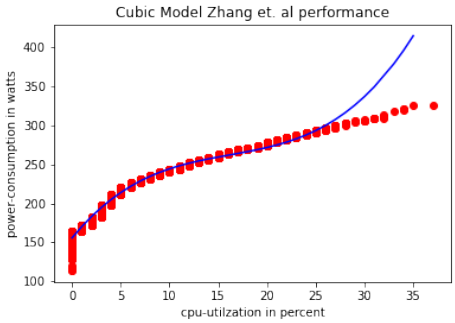
Several scientific articles note this problem with the R-squared metric like Spiess and Neumeyer [SN10] or Malkina-Pykh and Pykh [MPP19]. We include this value in our evaluation anyway, but the RMSE value might be a better basis for comparison between all four power models. For a comparison between 3.3 by Zhang et al. [ZLQZ13] and our polynomial model4.1 the R-squared value is still a good basis because these two equations are linear in their parameters. And generally, the trend that the R-squared value suggests is confirmed by the RMSE value in our evaluation even for the power model 3.2 of Fan et al. [FWB07] and our asymptotic model 4.2.



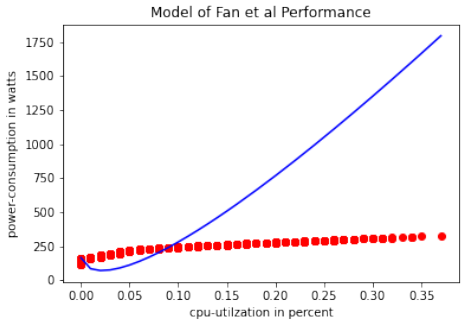
(a) Our Polynomial on XL450.



(b) Our Asymptotic power model on XL450.



(c) Power model 3.3 of Zhang et al. [ZLQZ13] on XL450.



(d) Power model 3.2 of Fan et al. [FWB07] on XL450.

Figure 7.8: The most inaccurate *calibrated maximum power* value.

8 Conclusion and Outlook

In this work, we train and analyse four server power models on a large dataset, which we obtain from a data center at the company AEB SE.

Two of those models are our contributions, the remaining two are from literature. The power models predicted in the following order from most accurate to least accurate: Our polynomial model 4.1, the power model 3.3 of Zhang et al. [ZLQZ13], our asymptotic model 4.2 and finally, power model 3.2 of Fan et al. [FWB07].

We find that our polynomial model shows the most promise in being generalisable. For this model, it seems to be an option to disregard storage size differences in categorising servers into configurations for training and even to disregard all differences. This does not seem to be the case for combining server types based on their CPU model though. Its speed of training is also comparatively good, on one level with the power model 3.3 of Zhang et al. [ZLQZ13]. Power model 3.3 of Zhang et al. [ZLQZ13] also shows promise to be generalisable, when it comes to storage size differences and is quite accurate. The other two models show less promise to be generalisable and should be trained with as fine-grained categories for training as possible in order to obtain accurate results.

In analysing and comparing the four models for their accuracy, generalisability, and their speed, we answer our research questions. The two new models that we introduce, especially our polynomial model 4.1, form another important contribution of this work.

With the dataset that we acquire here, we address a need for data from a real data center in research. In future work, this dataset can be analysed further by other researchers. It could, for example, be used to evaluate other power models.

CPU frequency, ambient temperature, and peak power are also included in the dataset and although we do not use these metrics for our power model evaluation, these could provide a basis for further research on server behaviour under real load.

Our finding that the power curves of the servers we analyse generally have the shape of a horizontal asymptote could be further analysed for its validity. Maybe this is just a feature of the HP Servers we analyse here and it does not apply to servers of other manufacturers.

If it does apply, it might be a good idea to refine the asymptotic power model we propose here. The shape of the curve is generally met by our model, but with some adjustments to the formula, a better fit might be possible.

Our idea of just increasing the degree of the cubic model that Zhang et al. [ZLQZ13] introduced to degree five could potentially be further explored in future work. Maybe the polynomial model 4.1 does not overfit the dataset yet and it is possible to increase the highest degree to seven or even higher. Since our polynomial power model 4.1 is quite accurate in many of the servers we analyse here, it might also be interesting to evaluate it further on servers from other manufacturers, since we only have access to data from servers of HP in this work and this might introduce a bias.

9 Acknowledgments

We want to thank the team at AEB for allowing us to get access to the data center and obtain the dataset that is the basis for this thesis. In particular, we would like to thank Markus W., Jens G., and Stephan Urbanski for their support and patience.

We would like to thank Brian Setz for giving us his support and guidance throughout the creation of this thesis.

Bibliography

[BVWS14] A. Barker, B. Varghese, J. S. Ward, I. Sommerville. "Academic Cloud Computing Research: Five Pitfalls and Five Opportunities". In: *6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14)*. Philadelphia, PA: USENIX Association, June 2014. URL: https://www.usenix.org/conference/hotcloud14/workshop-program/presentation/barker (cit. on p. 13).

[DWF16] M. Dayarathna, Y. Wen, R. Fan. "Data Center Energy Consumption Modeling: A Survey". In: *IEEE Communications Surveys Tutorials* 18.1 (2016), pp. 732–794. DOI: 10.1109/COMST.2015.2481183 (cit. on pp. 13, 19, 21).

[ERKR06] D. Economou, S. Rivoire, C. Kozyrakis, P. Ranganathan. "Full-system power analysis and modeling for server environments". In: International Symposium on Computer Architecture (IEEE). 2006. URL: https://scholarworks.calstate.edu/concern/publications/rr171x83v (cit. on p. 20).

[End] URL: https://techlibrary.hpe.com/docs/enterprise/servers/oneview5.0/cicf-api/en/index.html#rest/server-hardware?ref=GET%20%2Frest%2Fserver-hardware%2F%7Bid%7D%2Futilization&query=utilization (cit. on pp. 15, 25).

[FWB07] X. Fan, W.-D. Weber, L. A. Barroso. "Power Provisioning for a Warehouse-Sized Computer". In: *SIGARCH Comput. Archit. News* 35.2 (2007), 13–23. ISSN: 0163-5964. DOI: 10.1145/1273440.1250665. URL: https://doi.org/10.1145/1273440.1250665 (cit. on pp. 13, 20, 21, 24, 25, 32, 37, 38, 41, 45–51, 53–61, 63, 73–87).

[IM20] L. Ismail, H. Materwala. "Computing Server Power Modeling in a Data Center: Survey, Taxonomy, and Performance Evaluation". In: *ACM Comput. Surv.* 53.3 (2020). ISSN: 0360-0300. DOI: 10.1145/3390605. URL: https://doi.org/10.1145/3390605 (cit. on pp. 13, 19, 21, 22).

[Ilo] URL: https://support.hpe.com/hpesc/public/docDisplay?docLocale=en_US&docId=c04380351 (cit. on p. 15).

[Int] URL: https://github.com/scipy/scipy/issues/8439 (cit. on p. 60).

[MDS14] C. Möbius, W. Dargie, A. Schill. "Power Consumption Estimation Models for Processors, Virtual Machines, and Servers". In: *IEEE Transactions on Parallel and Distributed Systems* 25.6 (2014), pp. 1600–1614. DOI: 10.1109/TPDS.2013.183 (cit. on pp. 19, 21).

[MPP19] I. Malkina-Pykh, Y. Pykh. "Some notes on the application of R-squared for evaluation the goodness-of-fit of nonlinear regression models". In: *8th International Nonlinear Science Conference*. Coimbra, Portugal, Mar. 2019, p. 7. DOI: 10.13140/RG.2.2.35129.21607. URL: https://www.researchgate.net/publication/332207906_Some_notes_on_the_application_of_R-squared_for_evaluation_the_goodness-of-fit_of_nonlinear_regression_models (cit. on pp. 15, 16, 60).

[Man20] R. Manorathna. "k-fold cross-validation explained in plain English (For evaluating a model's performance and hyperparameter tuning)". In: (Dec. 2020). URL: https://www.researchgate.net/publication/348237224_k-fold_cross-validation_explained_in_plain_English_For_evaluating_a_model's_performance_and_hyperparameter_tuning (cit. on p. 17).

[McC+11] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, R. K. Gupta. "Evaluating the effectiveness of model-based power characterization". In: *USENIX Annual Technical Conf.* Vol. 20. 2011. URL: https://www.usenix.org/legacy/events/atc11/tech/final_files/McCullough.pdf (cit. on pp. 19, 20).

[Onea] URL: https://techlibrary.hpe.com/docs/enterprise/servers/oneview5.2/cicf-api/en/index.html#rest/server-hardware (cit. on pp. 24, 25, 57, 58).

[Oneb] *HPE OneView architectural advantages*. URL: https://www.hpe.com/us/en/collaterals/collateral.4aa5-3811.HPE-OneView-architectural-advantages-technical-white-paper.html?rpv=cpf&parentPage=/us/en/products/integrated-systems/management-software (cit. on p. 15).

[RRK08] S. Rivoire, P. Ranganathan, C. Kozyrakis. "A Comparison of High-Level Full-System Power Models". In: HotPower'08. San Diego, California: USENIX Association, 2008, p. 3. URL: https://www.usenix.org/legacy/events/hotpower08/tech/full_papers/rivoire/rivoire.pdf (cit. on pp. 20, 21).

[Rad+22] A. Radovanovic, B. Chen, S. Talukdar, B. Roy, A. Duarte, M. Shahbazi. "Power Modeling for Effective Datacenter Planning and Compute Management". In: *IEEE Transactions on Smart Grid* 13.2 (2022), pp. 1611–1621. DOI: 10.1109/TSG.2021.3125275 (cit. on pp. 13, 21).

[Rep] *Sklearn.modelselection.Repeatedkfold. scikit*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RepeatedKFold.html (cit. on p. 26).

[SN10] A.-N. Spiess, N. Neumeyer. "An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach". In: *BMC Pharmacology* 10.1 (June 2010), p. 6. ISSN: 1471-2210. DOI: 10.1186/1471-2210-10-6. URL: https://doi.org/10.1186/1471-2210-10-6 (cit. on p. 60).

[Scia] URL: https://scikit-learn.org/stable/index.html (cit. on p. 25).

[Scib] *3.3. metrics and scoring: Quantifying the quality of predictions*. URL: https://scikit-learn.org/stable/modules/model_evaluation.html#mean-squared-error (cit. on p. 16).

[Scic] *3.3. metrics and scoring: Quantifying the quality of predictions*. URL: https://scikit-learn.org/stable/modules/model_evaluation.html#r2-score (cit. on p. 15).

[Scid] $Scipy.optimize.curve_fit.scipy.optimize.curve_fit - SciPyv1.9.1Manual$. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html (cit. on pp. 24, 25, 53).

[Scie] *Sklearn.linear _model*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.score (cit. on p. 15).

[Scif] *Sklearn.linear _model*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.fit (cit. on pp. 25, 53).

[Spea] URL: https://spec.org/ (cit. on p. 17).

[Speb] URL: https://spec.org/power_ssj2008/results/ (cit. on p. 17).

[ZLQZ13] X. Zhang, J.-J. Lu, X. Qin, X.-N. Zhao. "A high-level energy consumption model for heterogeneous data centers". In: *Simulation Modelling Practice and Theory* 39 (2013). S.I.Energy efficiency in grids and clouds, pp. 41–55. ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2013.05.006. URL: https://www.sciencedirect.com/science/article/pii/S1569190X13000853 (cit. on pp. 16, 21–25, 31–33, 36–38, 41, 44, 45, 47–50, 52–55, 57–61, 63, 73–87).

All links were last followed on September 15, 2022.

A Table of unincluded Configurations

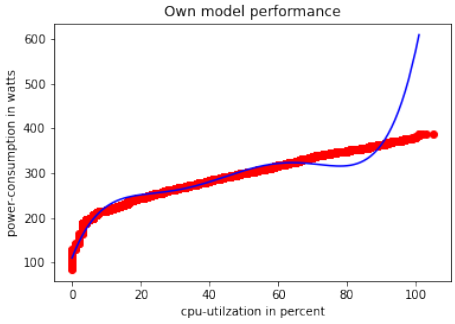
Here, we display the accuracy of the models on the server types that we have not included in our evaluation and discussion.

server-type	cpu-model	cpu-freq	core-count	server-count	r-sq	RMSE	power-model
SY480	X.G.+6342	2.8GHz	24	1	0.939	8.96	Zhang et al
SY480	X.G.+6342	2.8GHz	24	1	0.956	7.59	Own Poly
SY480	X.G.+6342	2.8GHz	24	1	0.755	17.93	Fan et al
SY480	X.G.+6342	2.8GHz	24	1	0.935	9.26	Asympt.
DL360p	X.E5-2640 0	2.5GHz	6	1	0.663	4.14	Own Poly
DL360p	X.E5-2640 0	2.5GHz	6	1	0.654	4.2	Zhang et al
DL360p	X.E5-2640 0	2.5GHz	6	1	0.247	6.2	Fan et al
DL360p	X.E5-2640 0	2.5GHz	6	1	0.521	4.94	Asympt.
DL380	X.E5-2630v3	2.4GHz	8	1	0.998	1.66	Own Poly
DL380	X.E5-2630v3	2.4GHz	8	1	0.988	4.28	Zhang et al
DL380	X.E5-2630v3	2.4GHz	8	1	0.902	12.13	Fan et al
DL380	X.E5-2630v3	2.4GHz	8	1	0.963	7.48	Asympt.
XL450	X.G.6230R	2.1GHz	26	1	0.982	2.77	Own Poly
XL450	X.G.6230R	2.1GHz	26	1	0.974	3.33	Zhang et al
XL450	X.G.6230R	2.1GHz	26	1	−34.511	122.6	Fan et al
XL450	X.G.6230R	2.1GHz	26	1	0.931	5.4	Asympt.
DL360	X.G.6128	3.4GHz	6	3	0.980	1.58	Zhang et al
DL360	X.G.6128	3.4GHz	6	3	0.981	1.55	Own Poly
DL360	X.G.6128	3.4GHz	6	3	0.977	1.69	Fan et al
DL360	X.G.6128	3.4GHz	6	3	0.974	1.8	Asympt.

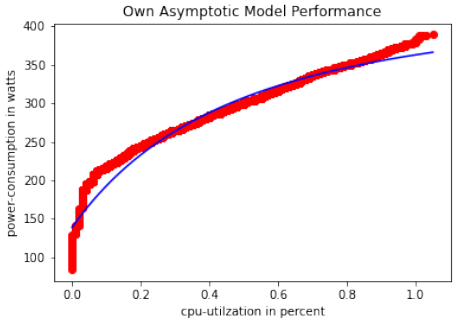
Table A.1: The accuracy of the power models on server configurations that we did not include already

B Unincluded Plots

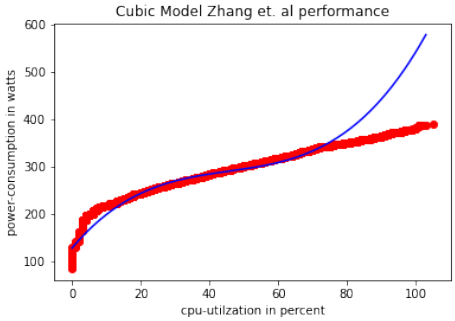
In this chapter of the Appendix, we shall include all plots, corresponding to the sections in the evaluation and discussion, that were not already included in the main paper.



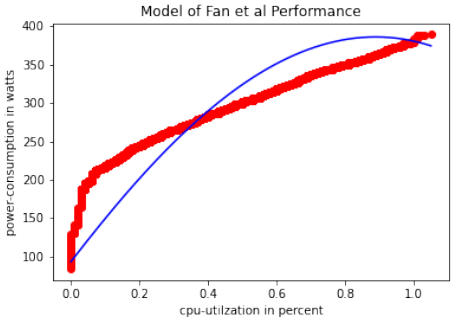
(a) Our Polynomial model.



(b) Our Asymptotic power model.

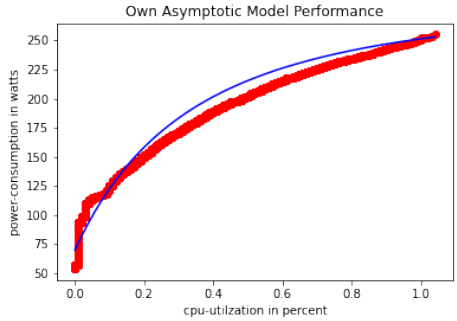


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

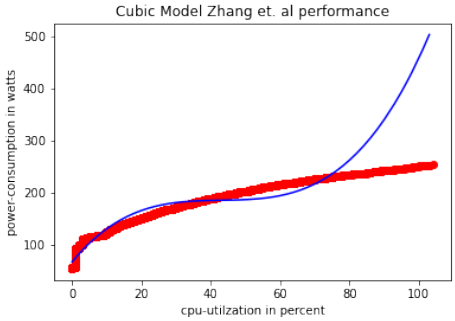


(d) Power model 3.2 of Fan et al. [FWB07].

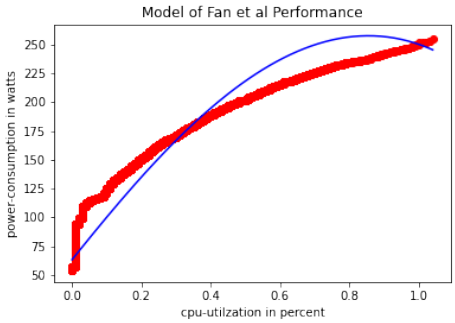
Figure B.1: Plots of BL460c CPU model Xeon E5-2660v3 without Storage Consideration.



(a) Our Asymptotic power model.

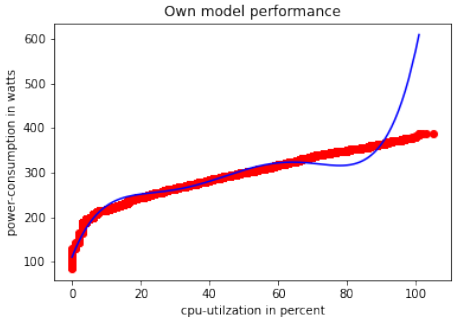


(b) Power model 3.3 of Zhang et al. [ZLQZ13].

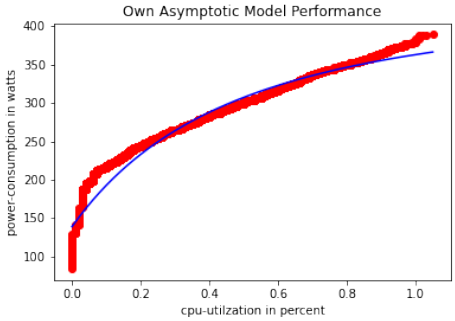


(c) Power model 3.2 of Fan et al. [FWB07].

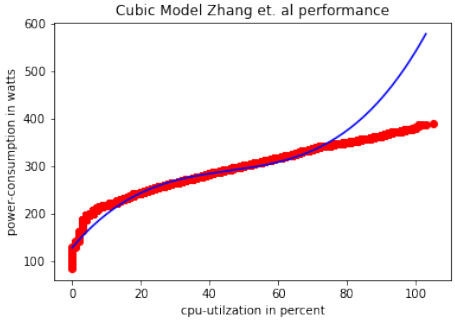
Figure B.2: Plots of BL460c CPU model Xeon E5-2640 without Storage Consideration, Own polynomial model included in Figure 6.8 on page 40.



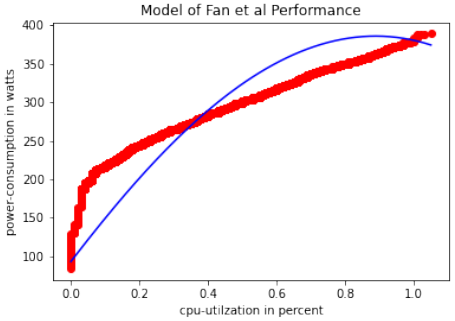
(a) Our Polynomial model.



(b) Our Asymptotic power model.

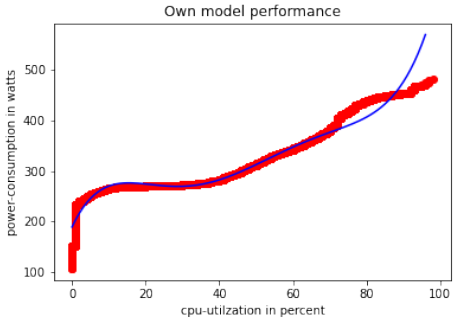


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

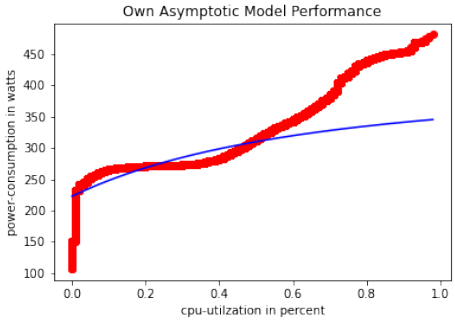


(d) Power model 3.2 of Fan et al. [FWB07].

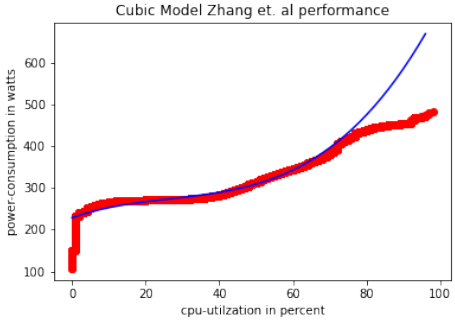
Figure B.3: Plots of BL460c CPU model Xeon E5-2660v3 without Storage Consideration.



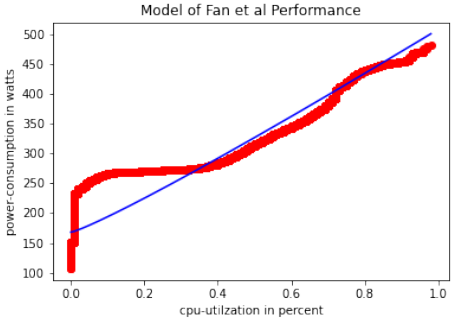
(a) Our Polynomial model.



(b) Our Asymptotic power model.

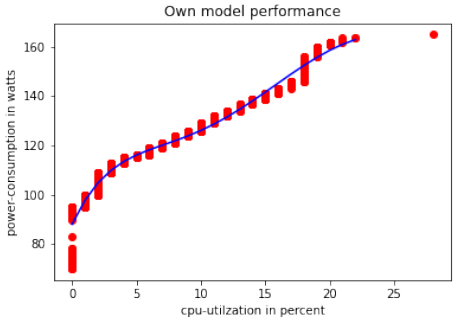


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

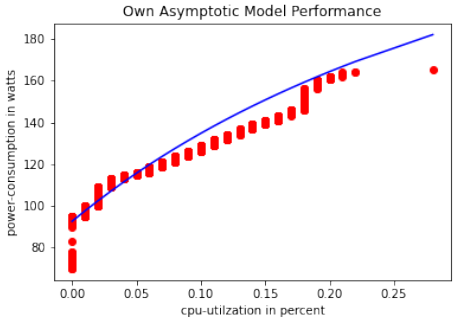


(d) Power model 3.2 of Fan et al. [FWB07].

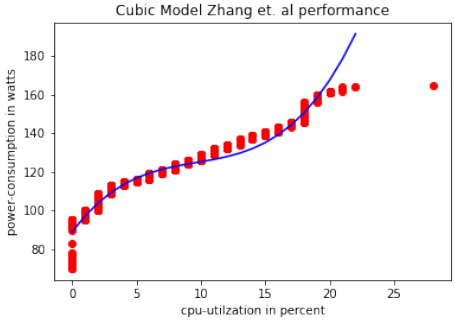
Figure B.4: Plots of SY480 with CPU model 6132 without Storage Consideration.



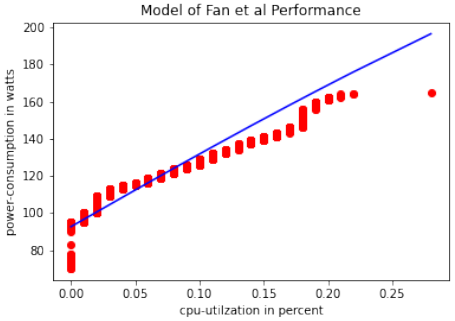
(a) Our Polynomial model.



(b) Our Asymptotic power model.

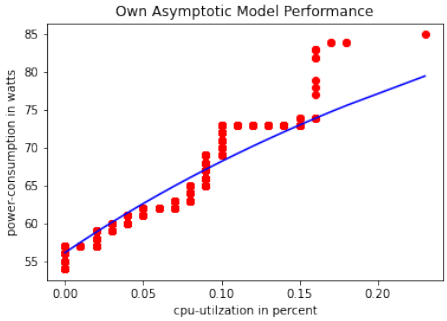


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

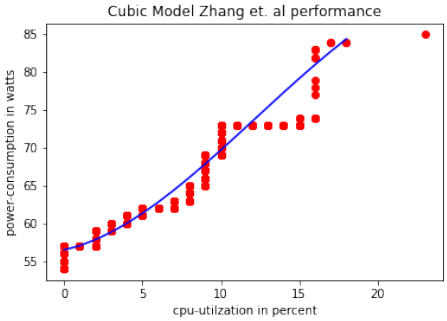


(d) Power model 3.2 of Fan et al. [FWB07].

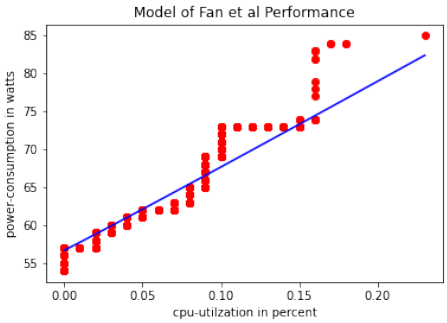
Figure B.5: Plots of BL460c CPU model Xeon E5-2640 with 64 GB Storage.



(a) Our Asymptotic power model.



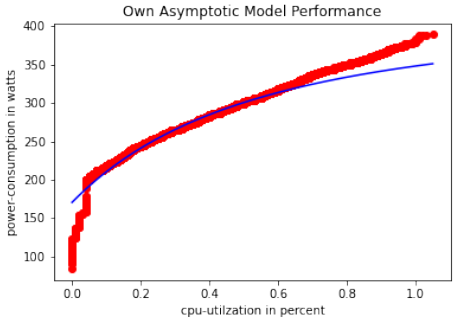
(b) Power model 3.3 of Zhang et al. [ZLQZ13].



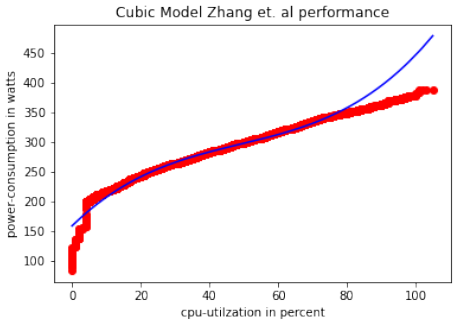
(c) Power model 3.2 of Fan et al. [FWB07].

Figure B.6: Plots of BL460c CPU model Xeon E5-2640 with 48 GB Storage. Figure of polynomial model included in Figure 6.8 on page 40

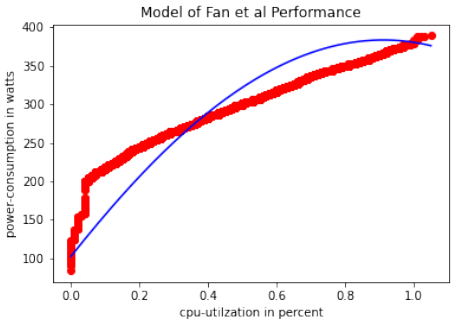
(a) Our Polynomial model.



(b) Our Asymptotic power model.

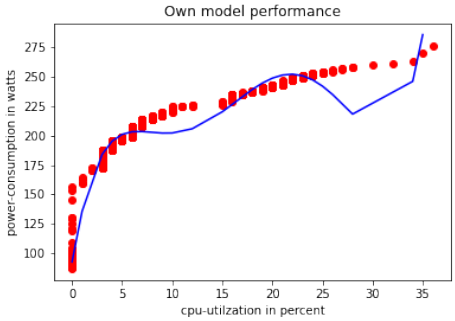


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

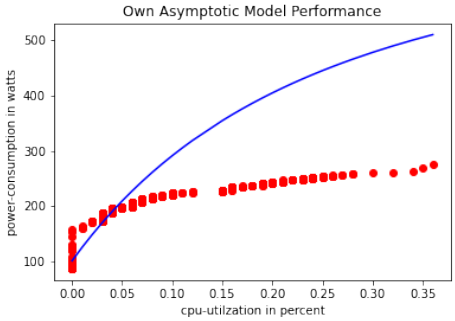


(d) Power model 3.2 of Fan et al. [FWB07].

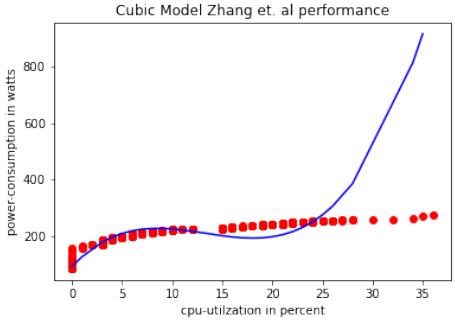
Figure B.7: Plots of BL460c CPU model Xeon E5-2660v3 with 1024 GB Storage.



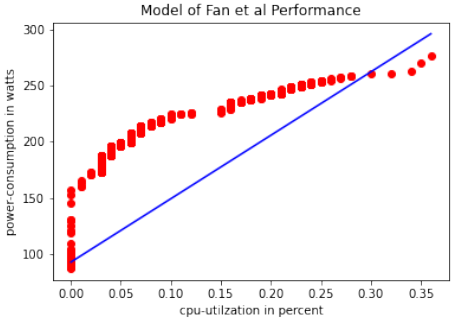
(a) Our Polynomial model.



(b) Our Asymptotic power model.

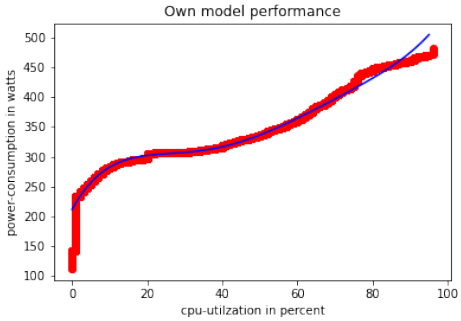


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

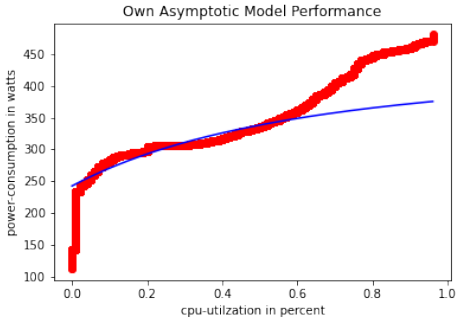


(d) Power model 3.2 of Fan et al. [FWB07].

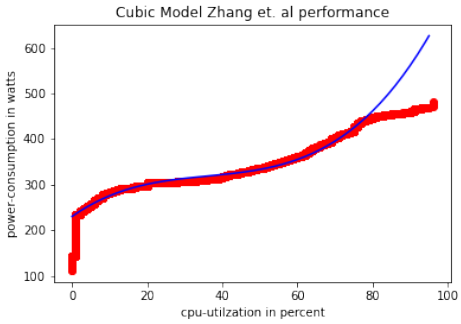
Figure B.8: Plots of BL460c CPU model Xeon E5-2660v3 with 512GB of Storage.



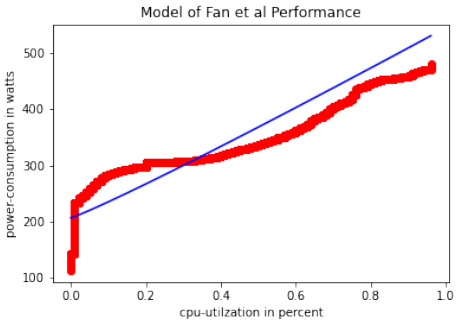
(a) Our Polynomial model.



(b) Our Asymptotic power model.

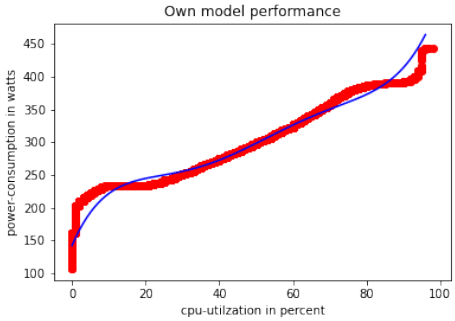


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

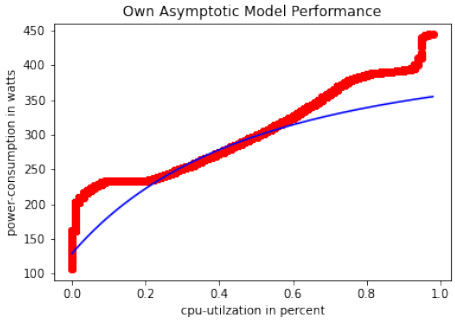


(d) Power model 3.2 of Fan et al. [FWB07].

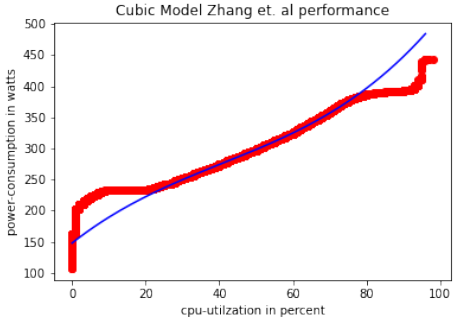
Figure B.9: Plots of SY480 CPU model Xeon Gold 6132 with 1536 GB Storage.



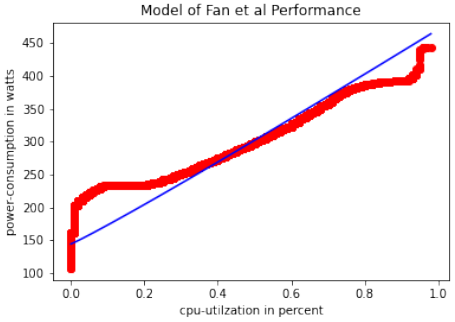
(a) Our Polynomial model.



(b) Our Asymptotic power model.

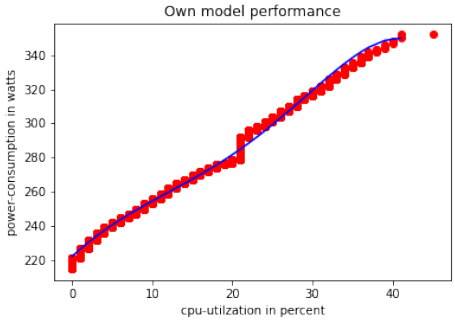


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

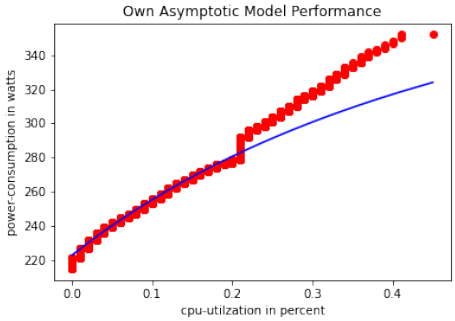


(d) Power model 3.2 of Fan et al. [FWB07].

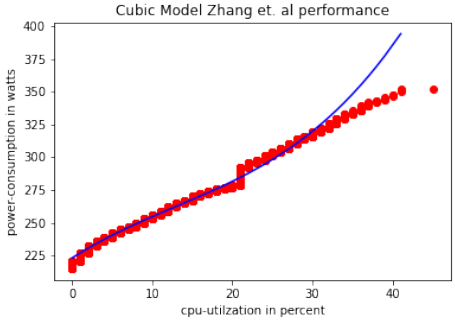
Figure B.10: Plots of SY480 CPU model Xeon Gold 6132 with 1024 GB Storage.



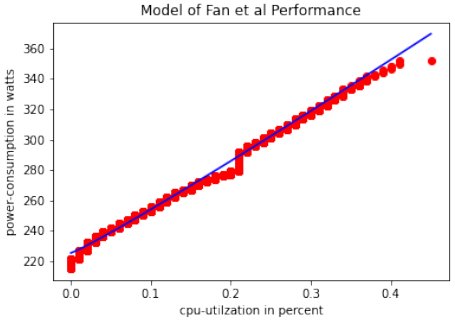
(a) Our Polynomial model.



(b) Our Asymptotic power model.

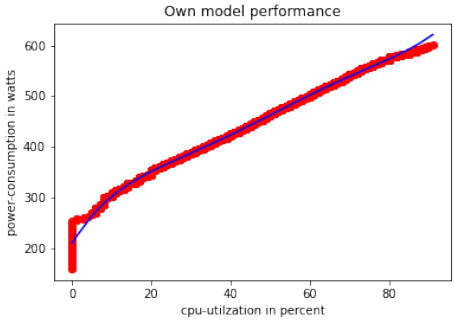


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

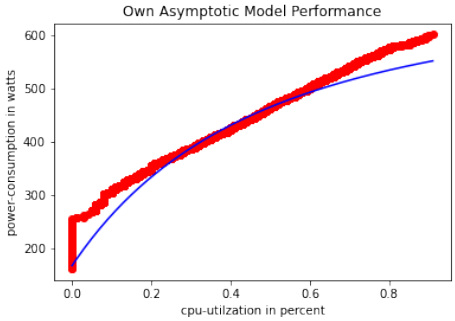


(d) Power model 3.2 of Fan et al. [FWB07].

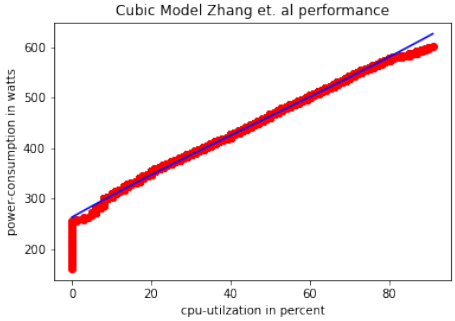
Figure B.11: Plots of SY480 CPU model Xeon Gold 6248 with 512GB Storage.



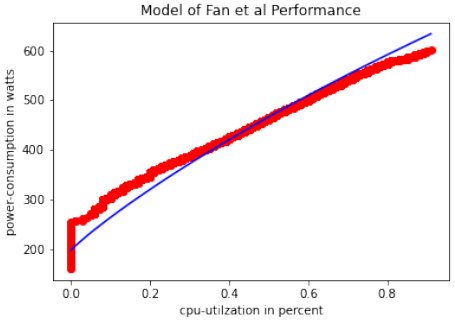
(a) Our Polynomial model.



(b) Our Asymptotic power model.

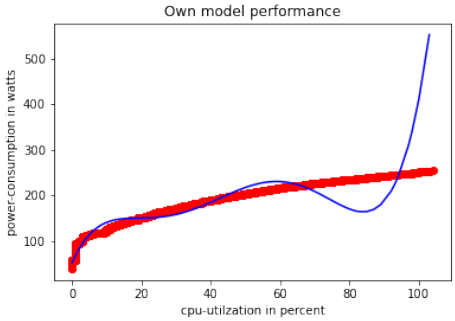


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

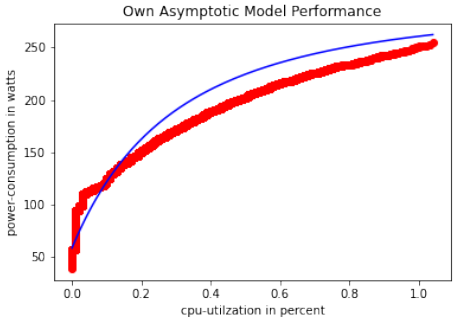


(d) Power model 3.2 of Fan et al. [FWB07].

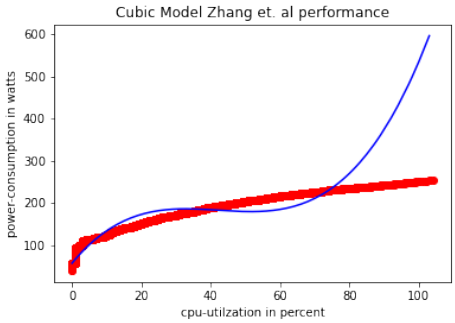
Figure B.12: Plots of SY480 CPU model Xeon Gold 6248 with 1536GB Storage.



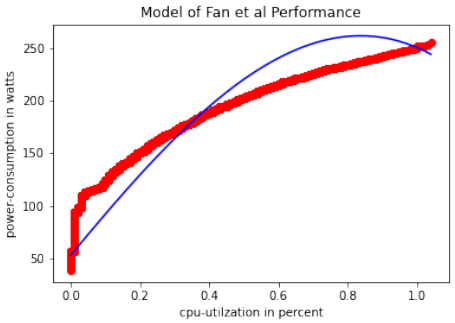
(a) Our Polynomial model.



(b) Our Asymptotic power model.



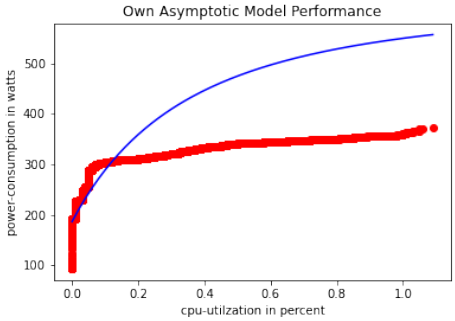
(c) Power model 3.3 of Zhang et al. [ZLQZ13].



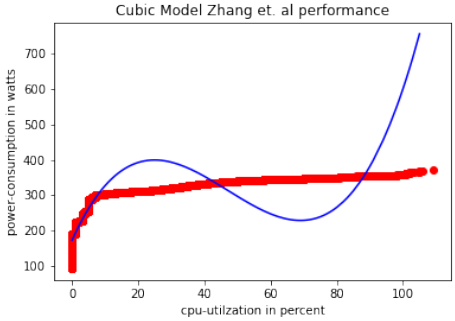
(d) Power model 3.2 of Fan et al. [FWB07].

Figure B.13: Plots of models on combined configuration for CPU model Xeon E5-2640.

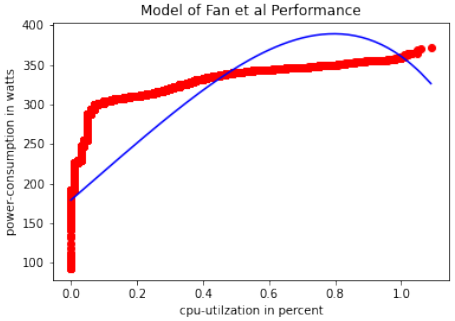
(a) Our Polynomial model.



(b) Our Asymptotic power model.

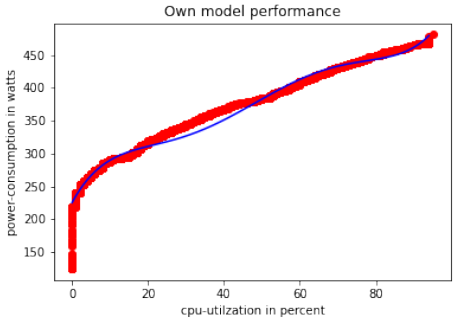


(c) Power model 3.3 of Zhang et al. [ZLQZ13].

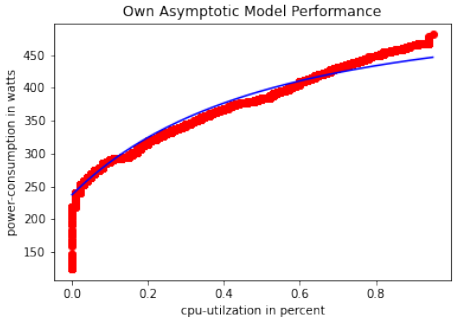


(d) Power model 3.2 of Fan et al. [FWB07].

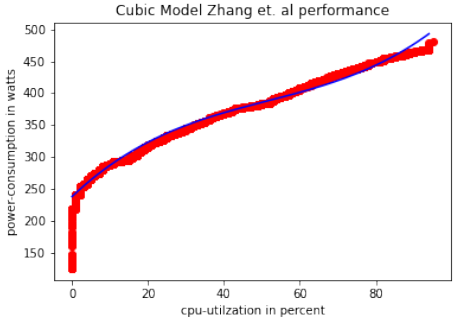
Figure B.14: Plots of models on combined configuration for CPU model Xeon E5-2690v4.



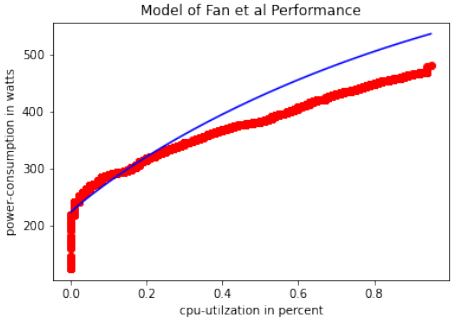
(a) Our Polynomial model.



(b) Our Asymptotic power model.



(c) Power model 3.3 of Zhang et al. [ZLQZ13].



(d) Power model 3.2 of Fan et al. [FWB07].

Figure B.15: Plots of models on configuration SY480 with CPU model Xeon Gold Plus 6342, 4096 GB of storage

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature