FACULTY OF TECHNOLOGY

Machine learning supported forecasting of baseline energy consumption for industrial processes

Olli Pesonen

DEGREE PROGRAMME IN PROCESS ENGINEERING

Master's thesis

December 2022

ABSTRACT

Machine learning supported forecasting of baseline energy consumption for industrial processes

Olli Pesonen

University of Oulu, Degree Programme of Process Engineering

Master's thesis 2022, 96 pp.

Supervisors at the university: Tero Vuolio, Markku Ohenoja, Mika Ruusunen

The purpose of the thesis was to study and evaluate machine learning supported methods in order to forecast a baseline energy consumption from time-series data of energy-intensive industry. In addition, time-series anomaly detection methods were studied and the anomaly detection accuracy of them was evaluated with hourly and daily average energy consumption data.

In the experimental part of the thesis a simulation scenario was established for hourly average data of two factories. The energy baseline was identified dynamically with week-ahead time-series forecasting by utilizing previous 52 weeks of data in the model training. In addition, model adaptation was considered in the simulation scenario. Predictor variables of the models were designed to imitate natural calendar effect. The energy baseline data of factory A was used to evaluate five linear and non-linear model structures. An average ensemble model structure appeared to outperform other model structures resulting in mean absolute percentage error of 9.3% for validation data of Factory A. The generalization ability of the model structure was evaluated with the data of factory B. For factory B the average ensemble model resulted in mean absolute percentage error of 9.9% for validation data. Overall, the results seemed promising especially as the set of input variables remained relatively simple as more precise subject matter expertise was not available during variable design and selection phase.

Keywords: energy baseline, machine learning, time-series forecasting, anomaly detection

TIIVISTELMÄ

Koneoppimiseen perustuva ominaisenergian kulutuksen ennustaminen teollisissa prosesseissa

Olli Pesonen

Oulun yliopisto, Prosessitekniikan tutkinto-ohjelma

Diplomityö 2022, 96 s.

Työn ohjaajat yliopistolla: Tero Vuolio, Markku Ohenoja, Mika Ruusunen

Diplomityön tavoitteena oli tutkia ja evaluoida koneoppimiseen pohjautuvia menetelmiä energiaintensiivisen teollisuuden aikasarjamuotoisen energiankulutusdatan käsittelyssä energiankulutuksen perusuran ennustamiseksi. Lisäksi työssä tutkittiin aikasarjadatan anomaliantunnistusmenetelmiä ja evaluoitiin niiden kykyä tunnistaa poikkeamia tunti- ja päiväkeskiarvoresoluutioisessa energiankulutusdatassa.

Työn kokeellisessa osiossa muodostettiin simulaatioskenaario kahden eri tehtaan vuosien 2020 sekä 2021 tuntikeskiarvoisten energiankulutusaineistojen mallinnukselle. Perusura muodostettiin dynaamisesti kerrallaan viikoksi eteenpäin aikasarjaennusteena edellisen 52 viikon aineistoa mallin opetuksessa hyödyntäen. Mallinnusskenaariossa huomioitiin lisäksi mallin suorituskyvylle olennainen adaptaatioproseduuri. Mallien selittävinä muuttujina käytettiin eksploratiivisen data-analyysin pohjalta luotuja luonnollista kalenterivaikutusta imitoivia muuttujia. Tehtaan A aineistolla evaluoitiin viittä eri lineaarista ja epälineaarista mallirakennetta. Parhaimmaksi mallirakenteeksi osoittautui keskiarvoyhdistelmämalli, jolle ennusteen keskimääräinen suhteellinen virhe oli 9,3 % validointiaineistolla. Mallirakenteen yleistyvyyttä testattiin toisen tehtaan (B) vastaavan ajanjakson aineistolla. Tehtaan B aineistolle keskiarvoyhdistelmämallin ennusteen keskimääräinen suhteellinen virhe oli 9,9 % validointiaineistolla. Tuloksia voidaan yleisesti ottaen pitää lupaavina etenkin, kun mallien tulomuuttujajoukko jäi verrattain yksinkertaiseksi, sillä tarkempaa aiheasiantuntemusta ei ollut saatavilla.

Asiasanat: energiankulutuksen perusura, koneoppiminen, aikasarjaennustaminen, anomaliantunnistus

FOREWORD

First and foremost, I would like to thank my employer ABB and the whole Digital Solutions team for giving me this opportunity and continuous support along the way. I want to thank my supervisor at ABB, Panu Karhu, for providing ideas and great tips when the subject and goals for the thesis were formulated. Also, special thanks to Juha Mäntysaari for your real interest and diligent advice in this project. In addition, thanks to my manager Kim Lampola and my mentor Ari Niinistö for supporting the thesis project from start to end.

Big thanks to University of Oulu. I am filled with gratitude for all the support I got from my supervisors Tero Vuolio, Markku Ohenoja, and Mika Ruusunen. Feedback was always constructive and I never felt that I was alone during this project.

I also want to give warm thanks for my family and others close.

Finally, I want to express my deepest gratitude to my dear grandfather, Alarik, who passed away shortly before the start of this project. I will never forget your encouraging words, and how much it meant for you that I will succeed in my studies.

Oulu, 14.12.2022

Olli Pesonen

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AR	autoregressive component
ARIMA	autoregressive integrated moving average model
ARMAX	autoregressive moving average model with exogenous variables
AS	anomaly score
CDD	cooling degree days
CNN	convolutional neural network
CUSUM	cumulative sum
EnB	energy baseline
EnPI	energy performance indicator
FFNN	feed-forward neural network
I	integrated component
iForest	isolation forest
ISO	International Organization for Standardization
iTrees	isolation trees
L_2	L_2-norm
LSTM	long short-term memory
MA	moving average component
MAE	mean absolute error
MAPE	mean absolute percentage error
MISO-MLP	multi-input single-output multi-layer perceptron
MLP-ANN	multi-layer perceptron artificial neural network
MLR	multiple linear regression
MSE	mean squared error
NARX	non-linear autoregressive model with exogenous variables
PCA	principal component analysis
ReLU	rectified linear unit
RMSE	root mean squared error
RNN	recurrent neural network
RSS	residual sum of squares
SARIMA	seasonal autoregressive integrated moving average model
SARIMAX	seas. autoregressive integrated mov. average model with exogen. variables
SARMA	seasonal autoregressive moving average model
SEC	specific energy consumption
SPC	statistical process control
SVM	support vector machine
SVR	support vector regression

1 INTRODUCTION

At the time of writing this thesis, the whole Europe is preparing for probably one the most challenging winters of this generation as the energy sector has drifted into turbulence. Because the electricity system must maintain the stability between supply and demand at any moment, enormous fluctuations are present on electricity prices as the portion of cheap renewable forms of energy, such as wind or solar, varies (Baev et al. 2022). For that reason, it is important for an organization to know their energy consumption characteristics to be able to cope with the fluctuations.

Energy baseline can be used for example to calculate the difference between the predicted consumption and the actual observed consumption, thus, to get metrics whether the energy consumption was on the level that the baseline model suggested. In addition, the forecast can be used to participate in the electricity markets as cost efficiently as possible. The baseline identification approaches were studied in terms of time-series forecasting methods in this thesis. The idea of this approach was to identify an energy baseline dynamically for the selected forecast horizon as a time-series forecast.

Secondly, anomaly detection is an important part of time-series analysis. The energy baseline itself can be used to detect anomalous energy consumptions as it provides a reference value where the energy consumption should lay given the conditions. In addition, various time-series anomaly detection methods can be utilized to find for example statistically abnormal data points. In this study various time-series anomaly detection methods were evaluated with hourly and daily average energy consumption data.

The expectations for the studies were to get an understanding which time-series forecasting approaches could work in situations where there is only the history data of the energy consumption and artificially designed variables imitating the effect of calendar available. Regarding time-series anomaly detection the interest was to get an idea which methods could be also effective and interpretable tools for the purpose.

The chapters apart from introduction are structured as follows. Chapter 2 gives an overview and explanation of the energy management in industry, and how it is linked to the ISO (International Organization for Standardization) 50001 and the ISO 50006 standards for energy management systems. Also, the key concepts, energy performance indicator and energy baseline, are described. Chapter 3 describes the key elements of time-series data and the key procedures when handling time-series data. The nature of time-series anomalies and some of the related methods are expressed in Chapter 4. Chapter 5 walks through the process of data-driven mathematical modelling. Used time-series forecast model structures are described together with information about variable selection, model parameter identification, and model validation and adaptation. Chapter 6 represents the materials and methods used in the experimental part of the thesis, and Chapter 7 explains the results. Finally, Chapter 8 concludes the thesis by discussing the results together with the findings from related literature and considerations for the future.

2 ENERGY MANAGEMENT IN INDUSTRY

An overview and explanation of the energy management in industry is provided in this chapter. Energy management is tightly linked to ISO 50001 and ISO 50006 standards in energy management systems and energy performance measurements, respectively.

Finnish national electricity transmission grid operator, Fingrid Oyj, published a network vision with different scenarios how the electricity system could shape in Finland in the upcoming decades (Fingrid Oyj 2022). It is certain that flexibility is a keyword in the puzzle of coping with the fluctuation in electricity market and prices. According to Fingrid Oyj (2022), this creates incentives particularly for demand side response and energy storage solutions. Basically, the electricity price is high when the portion of cheap renewable electricity sources, especially wind, is low and vice versa. When the portion is low, electricity is being produced with more expensive forms of production. Energy management solutions can be helpful in industry in order to adjust the demand, namely energy consumption, according to the cheaper hours while also considering the production schedule and its needs. Figure 1 illustrates the idea how the demand can be adjusted respect to time.

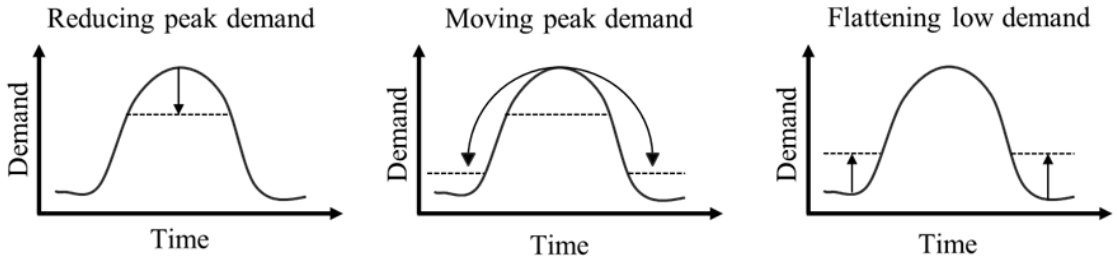

Figure 1. Examples of demand control.

2.1 ISO standards and energy baseline

The implementation of the ISO 50001 standard in energy management systems aims to enable companies and organizations to develop the needed systems and processes to improve energy performance in a continuous manner. The improvements in energy efficiency are often a consequence of improvements in the production processes and in

their energy management systems. According to Finnish Standards Association, energy performance consists of energy consumption, energy use and energy efficiency. (SFS-ISO 50001 2018)

The ISO 50001 delivery process considers commitment planning, implementation phase, checking, and reviewing. Commercial and academic tools are available for each phase. The commitment planning phase includes energy review and the identification of a baseline. The commitment planning phase is a continuous improvement process, in which the factors that affect the energy efficiency are identified. The reliable implementation of the commitment planning phase includes the verification of the improvement in energy efficiency by monitoring energy performance indicators. The baseline identification helps the organization to understand the consumption patterns in operation and to monitor the energy performance indicators. (Bruton, O'Donovan, McGregor & O'Sullivan 2018)

The ISO 50006 standard for energy performance indicators provides guidance for companies and organizations how to establish, maintain and use energy performance indicators (EnPI) and energy baselines (EnB). The defined metrics are used in the continuous energy performance measuring process related to energy management system implementations guided by the ISO 50001 standard. Finnish Standards Association describes an energy performance indicator as a ruler to compare the changes in energy performance before and after the energy performance improvements have been implemented. Figure 2 illustrates the idea of an energy performance indicator. The baseline period refers to the period before the improvement and reporting period refers to the period after the implementation. Also, targets are set for energy performance actions and therefore the current EnPI value reveals whether the targets are reached or not. (SFS-ISO 50001 2018)

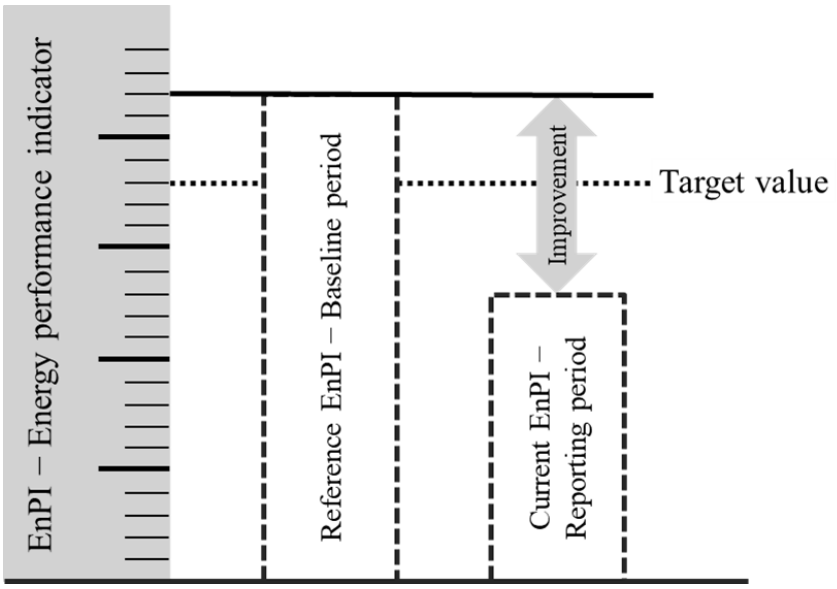

Figure 2. Energy performance indicator (retelling SFS-ISO 50001 2018).

Energy baseline is defined in the ISO 50001 and in the ISO 50006, and it is the quantitative basis for comparison of the energy performance. The baseline is identified based on the data from a specific time period and/or operating conditions. For distinct comparison, the organizations need to reliably identify energy baselines before and after the implementation of the designed energy performance improvement actions. The increase (or the decrease) of the energy performance defines the usefulness of the actions. Energy baseline must be identified for a suitable period of time which considers the operating cycles, the variables which have an effect on the energy consumption and the energy efficiency or regulatory requirements. It is mandatory that the selected data demonstrates the energy performance of the organization properly. The data used in baseline identification can be for example measured by the organization or queried from other sources for which the organization has access to, for example public meteorological data (SFS-ISO 50001 2018). In this study, the baseline for the upcoming week was identified dynamically with a week-ahead time-series forecast.

2.2 Definition and establishment of energy baseline

Measuring and collecting data has often been playing a keystone role in industrial development processes for a long time already. As more and more measurements are handled, preprocessed, and collected in industrial databases, the possibilities to utilize the data in various data engineering purposes increase. Although the amount of data is massive, the quality of the collected data must always be the number one priority. The concept *rubbish in, rubbish out* clarifies why it is essential to ensure the quality of the used data. If the quality is poor, not even the most complex mathematical modelling approach can save the situation. Measurements probably produce some kind of systematic bias, one way or another. Because of this together with the systematic variance of the modelling method, the results must be interpreted with *a grain of salt*. As George E.P. Box (1976) stated *"all models are wrong, but some of them are useful"*, even though it is more than expected that the modelling error is present, modelling can also be useful. Energy management makes no exception in this, as a systematic and data-driven approach in terms of energy baseline can be helpful to identify the possibilities how to handle the demand control situations. It is not only important in order to recognize the energy consumption characteristics of an organization, but also to make it easier to forecast the state in the future. It is cheaper for an organization to participate in the energy market well in advance with a clear view of their own electricity demand characteristics than having to fulfill the shortcomings in energy supply too late.

2.2.1 Model-based estimation and evaluation

There are many data-driven modeling techniques to identify a baseline model, for example statistical regression models and artificial neural networks, to name a few. The relevant variables for a baseline model can be for example weather data (indoor/outdoor temperature) and data purposefully derived from year calendar. Also, production volume together with planned and unplanned shutdowns play a crucial role in the energy consumption. Energy performance can be expressed with several different units: units of consumption such as GJ or kWh, specific energy consumption (SEC) such as

kWh/unit or peak power such as kW. The established energy baseline can be used to determine the difference between a baseline model and the observed energy consumption. (SFS-ISO 50006 2015)

The energy consumption pattern in process industry can be fairly complex due to possible high variation in production, especially in the product portfolio and in production rates. According to Hamedi & Mokhtar (2019) the commonly used energy performance indicator SEC might not be reliable enough in some cases because the number of influencing variables might be tremendous, and the variables are not taken into consideration well enough using SEC. Additionally, SEC might not include the constant part of the energy consumption, so called base load. (Hamedi & Mokhtar 2019)

A suitable period of time for the baseline needs to be considered carefully for each organization based on the nature of their operations. A one-year period is typically long enough to cover the full range of organization's business operating cycles, which are based on annual market demand patterns. Also, it is a practical period of time for doing the comparison regarding the energy consumption reduction. Energy baseline can also be identified for a shorter or longer period. A shorter period is suitable in cases where the shorter operation periods capture most of the operating patterns and if there are no seasonal changes in energy consumption. (SFS-ISO 50006 2015)

The validity of identified energy baseline model should be evaluated continuously by comparing the reporting period conditions to the baseline values of the selected variables. Also, in case of major changes made to the production process which could affect the energy consumption and performance, the baseline validity must be evaluated. Such major changes could be for example adding or deleting a production line or a change in production schedule by adjusting the shift calendar. From organization's perspective it might be useful to study the possible conditions in which the validity of the baseline should be re-evaluated. In addition, organization could determine the needed methods and procedures beforehand how the energy baseline should be adjusted. (SFS-ISO 50006 2015)

2.2.2 Establishment

Hamedi & Mokhtar (2019) studied the identification of an energy baseline for a low-density polyethylene production plant. Eight different independent variables were identified for the model, such as cooling degree days (CDD), number of plant's startups/shutdowns and the production rate. The dependent variables identified were the three consumptions: electrical energy, steam, and cooling water consumptions. Variable selection was done using engineering expertise from the plant. Multivariate linear regression (MLR) and multi-layer perceptron artificial neural network (MLP-ANN) were chosen as the studied model structures. The models were validated by using mean absolute percentage error (MAPE) as the metric. The energy baseline was defined as the model output predictions. The units for output predictions were MWh/day, ton/day and m^3/day for electrical energy consumption, steam consumption and cooling water consumption, respectively. The studies showed that both MLP-ANN and MLR models appeared robust, but the identified MLP-ANN model appeared fairly superior compared to the identified MLR model based on model accuracy. The studies suggest that more advanced artificial neural networks such as recurrent neural networks could be considered as the model structure for the energy baseline. (Hamedi & Mokhtar 2019)

3 THE ELEMENTS AND EXPLORATORY ANALYSIS OF TIME-SERIES DATA

To be able to describe time-series data it is important to understand the underlying patterns in it. The pattern types can be divided into four categories: trend, seasonal, cyclic, and stochastic patterns. The time-series data sets often include long-term trends, which are defined as periods of increasing or decreasing data values. The seasonal component is caused by seasonal factors, for example because of higher demand of heating energy during colder weather periods. The cyclic component is also following a constant pattern of increase and decrease but not in a fixed frequency like the seasonal component. The cyclic component could be better described for example with the economic conditions rather than calendar year. The stochastic component is then the remaining residual after extracting all the other parts from the data. The pattern types are illustrated in Figure 3. (Hyndman & Athanasopoulos 2018)

The forementioned patterns must be considered when dividing the data set for different purposes, for example to reselect it for training, test, and validation. Also correct

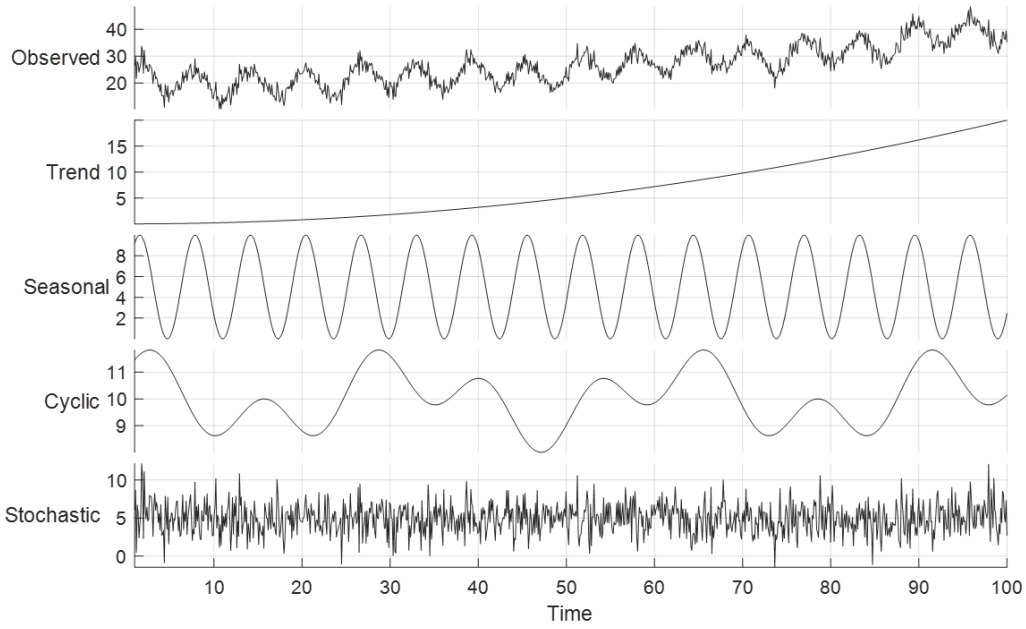

Figure 3. The four pattern types of a time-series data.

methods for cross-validation must be considered carefully when dealing with time-series data. In the time-series data the adjacent data points are not independent and therefore cross-validation procedure, if used, must be modified. (Arlot & Celisse 2010)

3.1 Data visualization and analysis

Exploratory data analysis is an integral part of data-driven modelling process. In this process, the essential information of the data is presented as summarizing statistical indicators, distributions, patterns, and correlations, to name a few. Visualization of the data as trend graphs and histograms is an efficient way to present time-series data (Komorowski, Marshall, Salciccioli & Crutain 2016). Usually if the data has a seasonal pattern, it is visible in both of those two. For example, a bimodal or a multimodal histogram is an indicator of multiple operating conditions with separate distributions.

Multimodality or existence of multiple operation points in the data is characteristic for industrial time-series data. For example, energy consumption might be much higher during a production day than during a day off. Also, depending on the production and shift calendar the energy consumption is probably higher daytime than it is during nighttime. Figure 4 clarifies six different and frequently present shapes of distribution within industrial time-series data.

Exploratory data analysis is an essential approach to identify different operating conditions. The operating conditions respect to time can be visualized for example using a two-dimensional heatmap where values in y-axis refer to the hour of the day and values in x-axis refer to the day of the week. To verify possible seasonal patterns in a time-series dataset, three important correlation metrics can be utilized: autocorrelation, partial autocorrelation, and cross-correlation. Autocorrelation and partial autocorrelation functions are also often used in the identification of a time-series model.

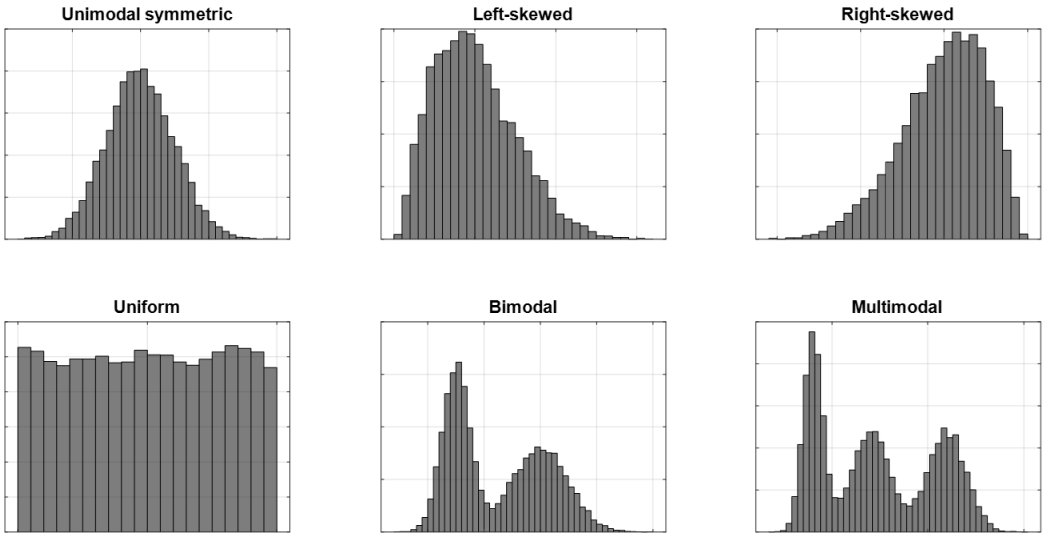

Figure 4. Different shapes of distributions.

Autocorrelation r_k for k^{th} lag is defined as follows (Box et al. 2015):

$$r_k = \frac{c_k}{c_0} = \frac{\frac{1}{N}\sum_{t=1}^{N-k}(z_t - \bar{z})(z_{t+k} - \bar{z})}{c_0}, \quad (1)$$

where c_k is the estimate of the autocovariance, $\bar{z}$ is the sample mean of the time-series, z_t is the value of the time-series at time t, N is the length of the time-series, and c_0 is the sample variance of the time-series. The standard error of the autocorrelation at lag k reduces to $\frac{1}{\sqrt{N}}$ if the time-series is totally random. Hence, the meaningful non-random lags can be identified. For seasonal time-series data the autocorrelation plot shows greater correlations for certain lags. For example, if the autocorrelation plot suggests that the hourly average variable correlates with itself offset by 24 hours, there is probably a 24-hour seasonal pattern. (Box et al. 2015)

Partial autocorrelation is also measuring the correlation between the time-series at time t and at time $t + k$. As an addition to autocorrelation, partial autocorrelation also adjusts for the linear effects of the time-series between time t and $t + k$. Cross-correlation function is used for multivariate time-series, and it is measuring the correlation between two different time-series with different lags. Cross-correlation

function can therefore reveal if a variable affects the future values of another variable. (Box et al. 2015)

3.2 Dataset shift

When dealing with any kind of prediction models an important assumption is that the training and test data joint distributions are similar, thus not dataset shifted. Usage of shifted test data may lead into wrong conclusions about the model's validity. A shift in the data may also indicate some underlying changes in the process. The change may go unnoticed if methodologies to identify these dataset shifts are unused. Quiñonero-Candela et al. (2009) divided dataset shifts in to six different types: simple covariate shift, prior probability shift, sample selection bias, imbalanced data, domain shift, and source component shift. Some of these types can be described shortly by their real-life reasons (Quiñonero-Candela et al. 2009):

- In **simple covariate shift** the distributions of covariates in test and training data are differing, meaning that the probability density functions of those two are inequal.
- In **prior probability shift** the distribution of the target changes.
- In **sample selection bias** the distributions are different because of an unknown rejection process of samples.
- In **domain shift** there is a change in the data distribution between the model training and model deployment. Therefore, the mapping of the covariates change between these two.
- In **source component shift** there is a change in the strengths of contributing components in the data.

One of the biggest issues with studying any type of dataset shifting is to determine if it exists. Quiñonero-Candela et al. (2009) continues by underlining that real practicalities should outweigh the theoretical details in some cases and highlights the need of testing the validity of a model in an actual real-life scenario before the model is finally implemented to be used. A modeling method which takes covariate shift into account

may perform worse than a standard model for data on which dataset shift does not occur. A potential procedure in energy management systems is to use dataset shift identification methodologies in order to spot the possible shifts. In case of a shift raise, an alarm triggers a deeper root cause analysis executed by subject matter experts. On the other hand, if there is enough precisely labeled data available, dataset shift classification can also be aided by machine learning techniques.

3.2.1 Measures of distribution similarity

The similarity of two observed distributions can be measured with different metrics such as Kullback-Leibler divergence (Kullback & Leibler 1951), Jaccard index (Jaccard 1912), and histogram intersection (Swain & Ballard 1991). Kullback-Leibler divergence, also known as relative entropy, between two probability distributions $P(x)$ and $Q(x)$ is calculated as follows (MacKay 2003):

$$D_{KL}(P\|Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)} \tag{2}$$

Jaccard index, also called as Jaccard similarity coefficient, is measuring the similarity of two distributions by dividing the size of the intersection by the size of the union. Jaccard (1912) originally formatted the index as follows:

$$Similarity = \frac{Number\ of\ observations\ in\ both\ sets}{Total\ number\ of\ observations} \tag{3}$$

This can be represented mathematically resulting in Jaccard index $J(A, B)$:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{4}$$

Histogram intersection measures the similarity of two discretized probability functions, namely histograms. The maximum value of histogram intersection is 1 when the distributions are identical, while a value of 0 refers to no overlap within the histograms.

The histogram intersection H_i for a pair of histograms I and M with n bins each, is calculated as follows (Swain & Ballard 1991):

$$H_i(I, M) = \sum_{j=1}^{n} \min(I_j, M_j). \quad (5)$$

3.2.2 Reasons for domain shift in industry

Harrou (2020) described the reasons and the consequences for faults in equipment or processes in industry. Common reasons are related to harsh operating conditions and aging of the equipment which usually lead to faults. According to Harrou (2020), a fault or anomaly is a bearable deviation of a variable from its actual acceptable behavior. A fault or anomaly can lead to a failure or malfunction if the fault is not detected and treated early enough. Harrou (2020) also underlines the difference between a failure and malfunction: in case of a failure the operation of a process is persistently suspended because process is unable to perform within its designated operating conditions, whereas malfunction causes irregular deviations in process's capability to perform its intended function. In industrial processes the underlying causes of anomalies can be for example faults or structural changes in processes, or faults in sensors and actuators. In terms of energy consumption, for example leakages in pipes could be an example of a significant process fault. Such leakages can be hard to be distinguished and are usually visible as slow changes across other related process variables. Faults in sensors and actuators can also be crucial and can occur in various ways. Harrou listed few examples of sensor faults such as bias errors, out of range errors, precision degradation errors, drift errors, and errors caused by a freezing sensor. Figure 5 illustrates various sensor faults (Harrou 2020). In Figure 5 the sensor faults demonstrate a situation which leads to domain shift, namely the measurements change so that they cannot be trusted or used in data-driven modelling.

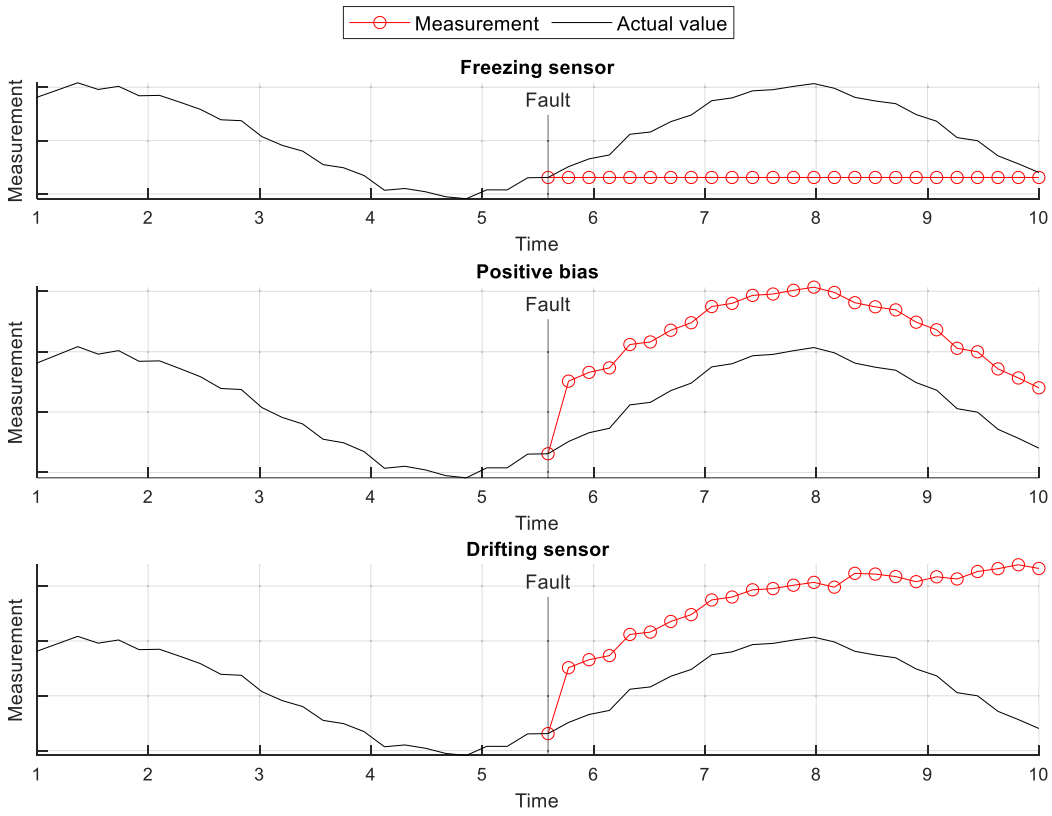

Figure 5. Different sensor fault types: freezing sensor, positive bias, and drifting sensor (retelling Harrou 2020).

4 TIME-SERIES ANOMALY DETECTION

In general, three types of anomalies occur in time-series data: point anomalies, pattern anomalies and sequence anomalies. Anomaly detection within time-series data can be executed with multiple different approaches; some being simpler and straight-forward, some being more complicated. Approach can be statistical based where the statistical distribution of the data is estimated and a comparison is done whether a measurement lies within the confidence interval or not, thus classifying the measurement as an anomaly or not. Statistical based method can be executed for example in a rolling window where distribution of the measurements of recent past is estimated and the newest measurement is compared to the distribution. (Wang et al. 2022)

4.1 Three-sigma rule

The so-called three-sigma rule of thumb represents a situation in which the outcome of a process follows normal (Gaussian) distribution, and thus it is expected that 99.73% of the measurement population lays within the range of ± 3 standard deviations (σ) around the mean of the population. According to Oakland (2003) a process is considered to be in statistical control if all the occurring variation can be shown to appear due to random or common causes. Oakland states that if the process is in statistical control there should not be for example any measurements outside the ± 3σ area, unusual trends, or multiple consecutive measurements on the same side of the population mean. There are many widely utilized rules to detect the special causes which initiate unwanted variation in a process. The most widespread rules are demonstrated by Western Electric (1956) and by Nelson (1984) and (1985). The idea behind the rules can be best illustrated with a control chart and with the corresponding zones in it (Figure 6).

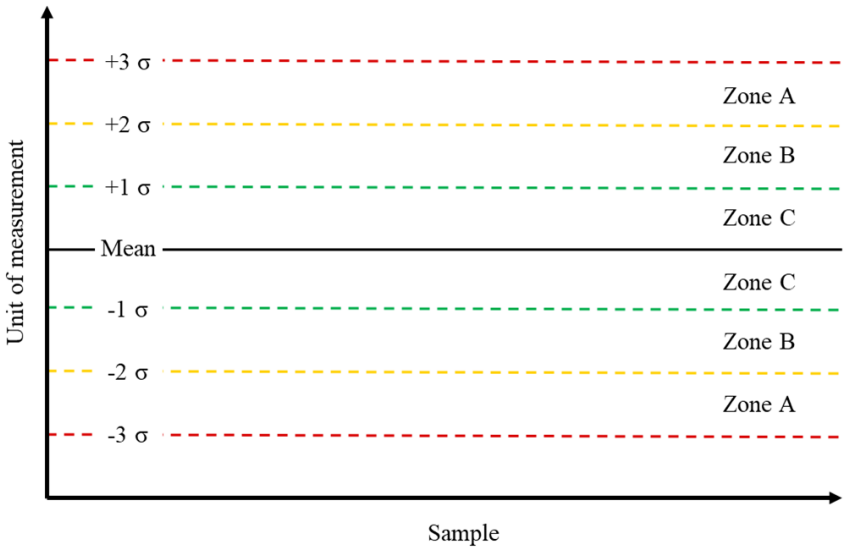

Figure 6. Illustration of a control chart (retelling Oakland 2003).

In terms of anomalies in energy consumption the most unwanted situations occur when the consumption is higher than it should be, or if the consumption is slowly drifting upwards. Therefore, in this study the focus on the selected rules for statistical process control purposes is on the deviations above the population mean, and on the occurring unusual trends. On the other hand, also the periods when the consumption is shifted below the population mean might be interesting. By spotting the lower energy consumption periods and analyzing the cause for the lower energy consumption itself may lead into improvements and better practices in the operations.

4.2 Isolation forest

Isolation forest (or *iForest*) is an anomaly detection algorithm proposed by Liu et al. (2008) and it is based on an ensemble of isolation trees (*iTrees*). With term *isolation* the authors mean "separating an instance from rest of the instances", and in the algorithm the anomalies which are "few and different" are isolated from other data points in the branches of iTrees. The algorithm proceeds by dividing the data randomly and recursively until the height limit of the tree is reached or the data in one node of the tree have the same values. Each data point is ranked based on an anomaly score and on the path length to reach the point in the tree. Figure 7 illustrates the structure of an isolation forest. Green color indicates a path from a parent node of an isolation tree to a sample identified as common measurement. (Liu et al. 2008)

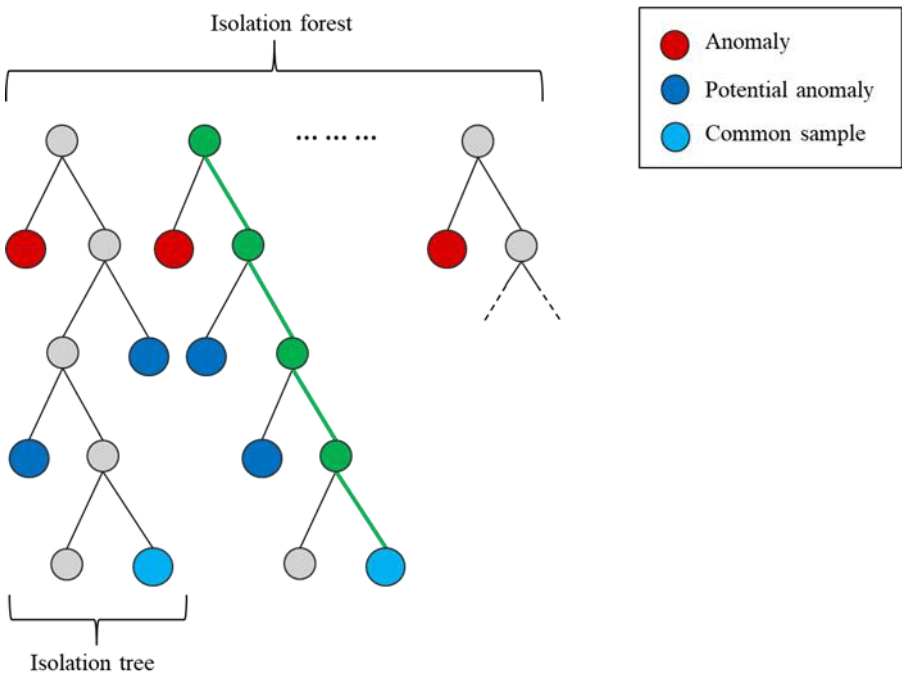

Figure 7. Structure of an isolation forest (retelling Liu et al. 2008).

4.3 Autoencoder networks in anomaly detection

Autoencoder is a machine learning approach for converting high-dimensional data into low-dimensional code and recovering the data with a decoder. To simplify, the system as a whole is a multilayer network which has a structure kind of an hourglass. It consists of three parts: encoder, bottleneck, and decoder. The encoder part takes the input, and the input is squished to lower dimension layer-by-layer until it reaches the bottleneck. After the bottleneck the input is upscaled again layer-by-layer on the decoder part until it reaches the output layer. Therefore, the output of the network is a reconstruction of the input. The method is used for example to reduce the dimensionality of data. Conventional methods such as principal component analysis (PCA) are simpler and probably more widely used but according to Hinton & Salakhutdinov (2006) a well-trained deep autoencoder can outperform PCA with a much better reconstruction error. A simplified structure of an autoencoder network is on Figure 8.

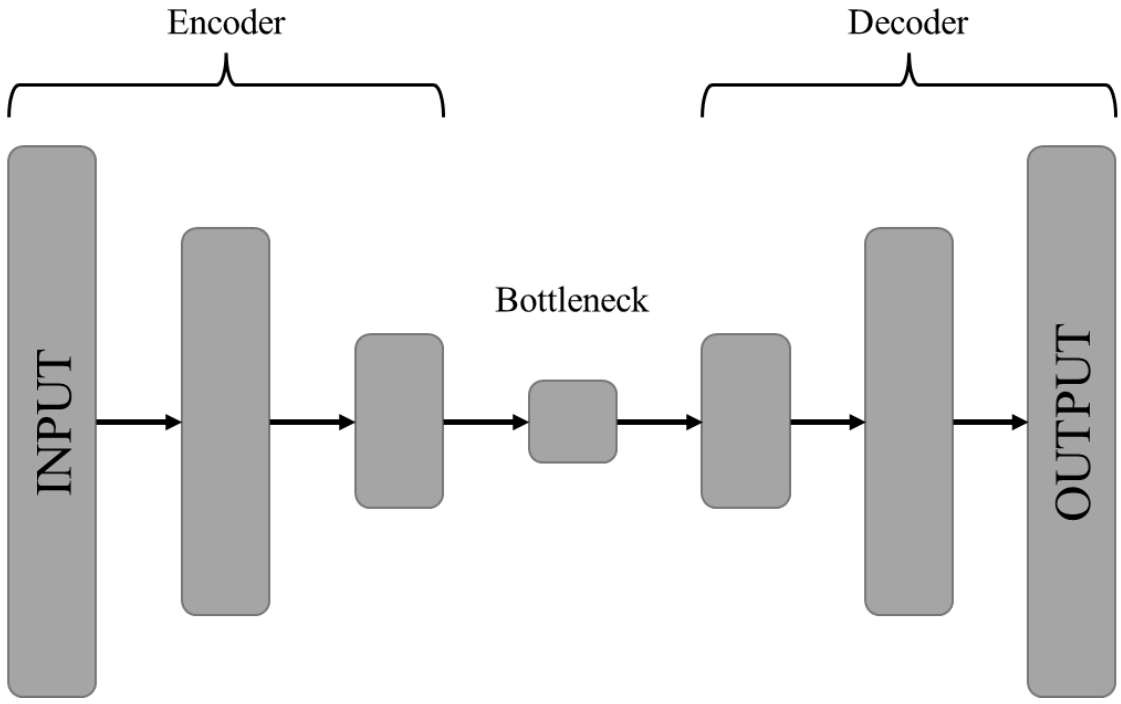

Figure 8. Simplified structure of an autoencoder network.

The idea behind using autoencoder as an anomaly detection method is on the reconstruction error for the input data. As the autoencoder model is trained with some history data of the process, it will learn how to reproduce that exactly same data. Let's say we have trained the autoencoder with data captured during normal operating conditions. When the data captured in the future is fed in to the autoencoder network, it should be reproduced according to the training data if the normal operating conditions are still present. If the future data happens to be anomalous, the autoencoder cannot reproduce the data and it will be visible as reconstruction error. The reconstruction error of point i can then be transformed into some anomaly score metric. A threshold is then set for the anomaly score, and the threshold is used to decide whether a point is an anomaly or not.

5 MODELLING OF ENERGY BASELINE

This chapter provides an overview of popular techniques related to time-series forecasting used in this study.

5.1 Supervised learning

According to James et al. (2013) machine learning problems are usually either supervised or unsupervised. In supervised learning, the training set consists of values of input variables with the corresponding labeled output values. The function f which maps the input x to the output $f(x)$ so that the error ε is as small as possible, is attempted to be identified in supervised learning. In unsupervised learning there is no associated response available and thus the response cannot supervise the learning outcome. The energy baseline identification related regression problems studied in this thesis fall under the category of supervised learning.

After model structure selection and variable selection, the function identification is about model parameter identification. This two-step model-based approach is also called *parametric learning* (James et al. 2013). For parameter identification there are multiple approaches, selected subject to problem complexity. For regression problems the cost function of the problem is typically sum-of-squared errors $R(\beta)$, (Hastie et al. 2009):

$$R(\beta) = \sum_{k=1}^{K} \sum_{i=1}^{N} \left(y_{ik} - f_k(x_i)\right)^2, \tag{6}$$

where **y** is a labeled output and $f(x)$ is a function of input **x** with weight parameters β. Therefore, the model fitting is about searching for the optimal values of the weight parameters so that the sum-of-squared errors gets its minimum value. For linear regression there is a unique solution available for weight parameters, which is explained in Chapter 5.2.1. For non-linear regression problems such as neural networks the usual

optimization algorithm is gradient descent based. One usually applied stochastic gradient descent algorithm is *Adam* (Kingma & Ba 2014) which can handle complex problems with stochastic objective functions.

5.2 Model structure

Choosing an appropriate model structure plays a key role in mathematical modelling. The wide variety of different model structures gives lots of possibilities to work with but may also cause issues if the model structure is not selected properly (James et al. 2013). The output type of a model in a prediction task is typically categorized into two classes: regression and classification. Regression refers to a task in which the output is quantitative, whereas classification refers to a task in which the output is qualitative. In this study both regression and classification models are studied since time-series forecasting is a regression task whilst anomaly detection is a classification task. The focus in this chapter is on regression models. (Hastie et al. 2009)

Bias-variance tradeoff (illustrated in Figure 9) is an important concept in model structure selection. According to Hastie et al. (2009) the complexity of a model is directly proportional to variance and inversely proportional to squared value of bias.

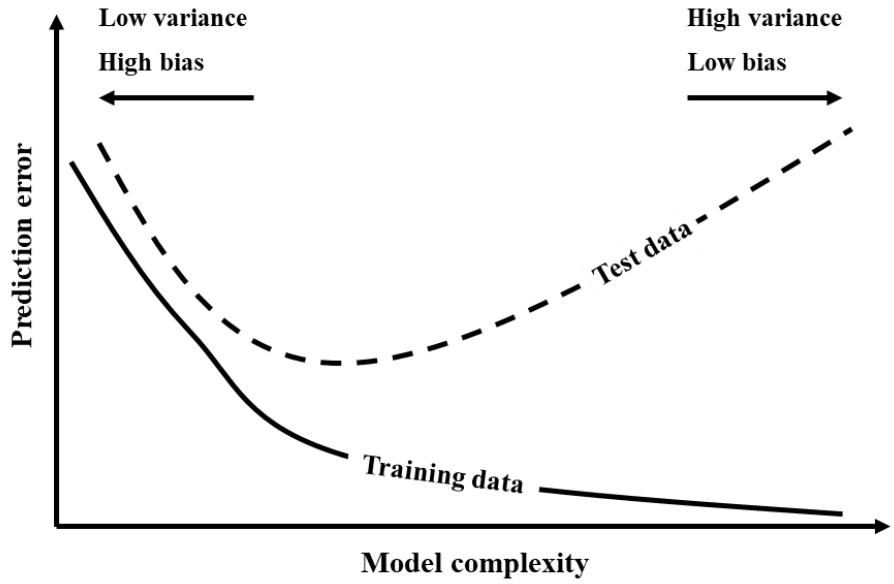

Figure 9. Model complexity versus prediction error for training and test data (retelling Hastie et al. 2009).

This means that when the model complexity is increasing, the error of the output tends to decrease. On the other hand, if the model complexity gets too high the situation leads into overfitting because the model predicts the training data too well. A model with tendency to overfitting does not generalize well for test data and will result in large error. Similarly, a model which is not complex enough has tendency to underfit. An underfitted model has a large bias and again the result is a poor generalization ability.

5.2.1 Simple linear models

One of the simplest approaches is the multivariable linear regression which is a predictor based on linear mapping (Hastie et al. 2009):

$$y = X\beta + \varepsilon, \tag{7}$$

where $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_n]^T$ is a real-valued output vector (size $n \times 1$), $\mathbf{X} = [1 \ \mathrm{x}_1 \ \dots \ \mathrm{x}_n]$ is an input matrix with input vectors ($\mathbf{x_n}$) as columns ($n \times p$), $\boldsymbol{\beta} = [\beta_0 \ \beta_1 \ \dots \ \beta_n]^T$ is the unknown regression model parameter vector ($p \times 1$), and $\boldsymbol{\varepsilon} = [\varepsilon_1 \ \varepsilon_2 \ \dots \ \varepsilon_n]$ is the error vector ($n \times 1$). The first column vector of the input matrix is a vector of ones for adding the bias term β_0 for each output prediction. Regarding matrix sizes, n refers to the number of data points and p refers to the number of regression parameter terms, thus, the number of input variables is $(p - 1)$.

There exists a unique solution for vector $\widehat{\boldsymbol{\beta}}$ if input matrix $\mathbf{X}$ has full column rank, namely the columns of $\mathbf{X}$ are linearly independent, and therefore matrix multiplication $\mathbf{X^T X}$ is positive definite. By differentiating the sum of squared residuals $R(\beta)$ (Equation 6) with respect to $\boldsymbol{\beta}$ and by setting the first derivative to zero the unique solution can be obtained as follows (Hastie et al. 2009):

$$\hat{\beta} = (X^T X)^{-1} X^T Y. \tag{8}$$

5.2.2 Linear time-series models

A very popular stochastic time series model Autoregressive Integrated Moving Average (ARIMA) is based on two assumptions: the studied time series is linear, and it follows a statistical distribution which is known, for example the normal distribution. ARIMA model consists of autoregressive (AR), integrated (I), and moving average (MA) components. Autoregressive and moving average components can be used as separate model structures, or together without the integrated component, based on the studied data. When modeling a dataset with seasonal component, also a seasonal component (S) can be added to ARIMA model resulting as so-called SARIMA model. In the event that the integrated component is left out, a SARMA model is obtained. If other variables influence the forecasted time series, they can be included in the model as exogenous terms (X), resulting as so called SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Variables) model. The $SARIMAX(p,d,q)(P,D,Q)_S$ model at time t with multiple exogenous variables in vector $\mathbf{X_i}$ is obtained as follows (Nontapa et al. 2020):

$$y_t = \beta_0 + \sum_{i=1}^{k} \beta_i X_{i,t} + \frac{\theta_q(B)\Theta_Q(B^S)}{\phi_p(B) + \Phi_P(B^S)(1-B)^d(1-B^S)^D}\varepsilon_t\,, \tag{9}$$

where y_t is the observation at time t, ε_t is random error at time t, β_0 is the constant parameter of multiple linear regression, $\boldsymbol{\beta_i}$ is a vector of regression coefficient parameters of exogenous variables, $\mathbf{X_{i,t}}$ is a vector of observations of exogenous variables at time t, $\phi_p(B)$ is the non-seasonal AR operator of order p, $\Phi_P(B^S)$ is the seasonal AR operator of order P, $\theta_q(B)$ is the non-seasonal MA operator of order q, $\Theta_Q(B^S)$ is the seasonal MA operator of order Q, S is the length of the season, and d and D refer to the order of the differencing operator I.

5.2.3 Machine learning and deep learning models

Deep learning methods is a sub-group of computational machine learning models which are structured to process the input data in multiple layers in order to learn the patterns in

the data in multiple levels of abstraction. As in machine learning overall, supervised learning is the most common form of deep learning. Whereas conventional techniques of machine learning may struggle to process data in its natural, raw form, deep learning techniques can reduce the need for careful engineering to pre-process the raw data, for example to extract the features. (LeCun et al. 2015)

Feed-forward neural networks

One of the simplest and most common structures of feed-forward neural networks (FFNN) is the multilayer perceptron (MLP) which consists of three fully connected layers: input layer, hidden layer, and output layer. The hidden layer itself can then have one or more layers. (LeCun et al. 2015)

Figure 10 represents a fully connected feed-forward multi-input-single-output (MISO) multilayer perceptron with four neurons on both of the two hidden layers. The computational unit of a neural network is called neuron. Full connection means that every neuron of a layer is connected to each neuron on the previous layer and on the following layer. The connections are given a weight and the connections coming to a neuron are added up together with the bias of the neuron. Thus, the total input of a

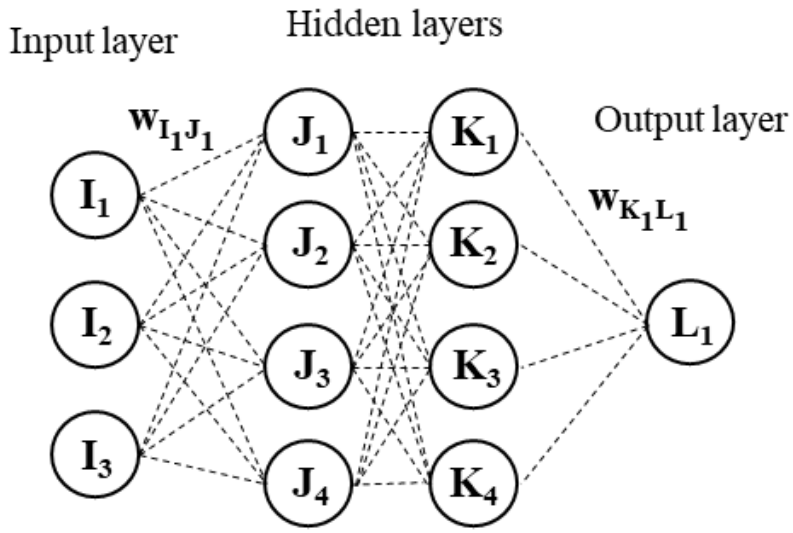

Figure 10. MISO-MLP with two hidden layers (retelling LeCun et al. 2015).

neuron is computed as a linear map (LeCun et al. 2015):

$$z = wx + b, \tag{10}$$

where z is the total input of the neuron, **x** is the input vector, **w** is a vector with weight parameters for each corresponding input and b is the bias of the neuron. The total input is then fed to a non-linear function f such as rectified linear unit (ReLU) or some sigmoid function, for example hyperbolic tangent (tanh), in order to compute the output of the neuron (LeCun et al. 2015):

$$y = f(z). \tag{11}$$

The output of the neuron is then weighted with a weight parameter and connected to every neuron of the following layer. Figure 11 illustrates the flow of information through a neuron in MLP. An MLP is usually trained using the backpropagation procedure in which the chain rule of derivatives is applied to adjust the weights of the connections in order to find an optimum of the objective function. In backpropagation the output is calculated based on the inputs and the error of the output is then passed back to the input layer. The weight parameters of the MLP are tuned simultaneously from end to start, namely from output to input layer, intending to minimize the

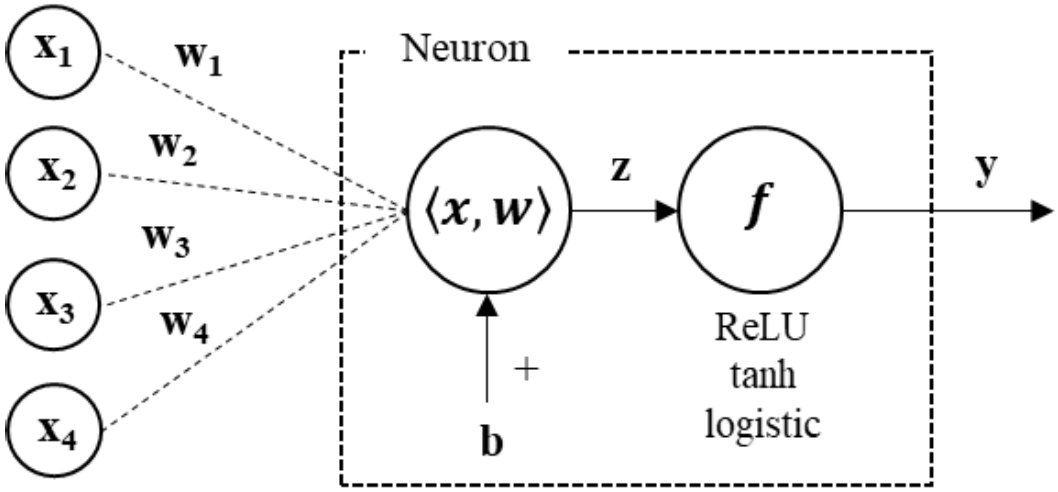

Figure 11. Information flow through a neuron in MLP (retelling LeCun et al. 2015).

modelling error. (LeCun et al. 2015)

Convolutional neural networks

In case of time-series data a convolution can be seen as sliding a filter over the data. The result of the convolution applied on a univariate time-series is given as follows (Ismail et al. 2019):

$$C_t = f\left(\langle \omega, x_{t-\frac{l}{2}:t+\frac{l}{2}} \rangle + b\right) \;\Big|\; \forall\, t \in [1, T]\,, \tag{12}$$

where $\mathbf{x}$ is the univariate time-series of length T, $\boldsymbol{\omega}$ is the filter of length l, b is the bias term and f is the non-linear function, such as ReLU, applied to the dot product. So that the filters are invariant of time, the same filter values $\boldsymbol{\omega}$ and b will be used for every timestamp $t \in [1, T]$. After the convolution local pooling, either average or max pooling, is applied to the input time-series. In pooling the length of the time-series is reduced depending on the length of the sliding window. Lastly, the result of the convolutions is the representation of the input time series for a feed-forward neural network, namely the inputs to the FFNN. Therefore, in short convolutional neural networks (CNN) are a combination of convolutional operations and fully connected neural networks. (Ismail et al. 2019)

Recurrent neural networks and long short-term memory

Recurrent neural networks (RNN) might come in handy with sequential data, as in this case with time-series data acquired from energy-intensive industry. The main idea of RNN is to process the input sequence part by part while also considering the information of the parts in the past. An illustration of the general structure of an RNN and how the structure looks unfolded can be seen on Figure 12.

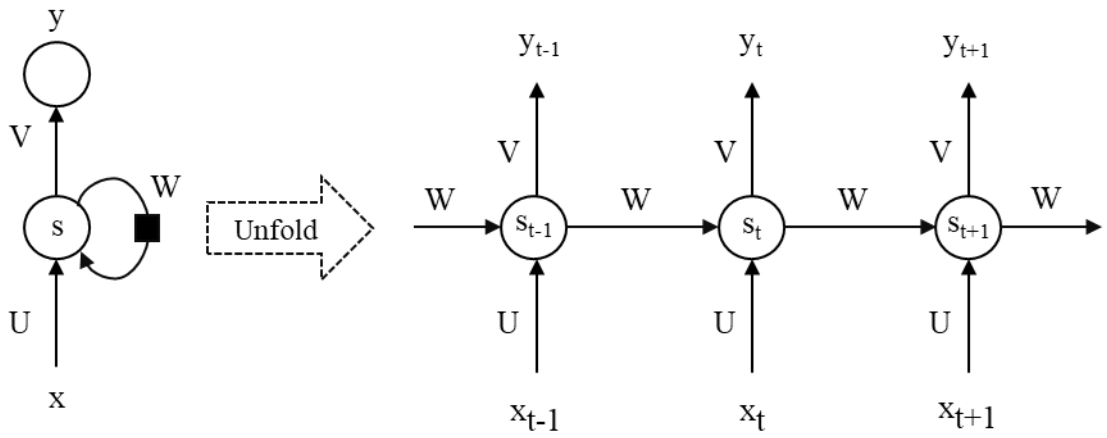

Figure 12. Recurrent neural network (retelling LeCun et al. 2015)**.**

In Figure 12 $\mathbf{x}_t$ refers to the input sequence at time t, s refers to the node with values $\mathbf{s}_t$ at time t and $\mathbf{y}_t$ refers to the output sequence at time t. Parameter matrices (**U**, **V** and **W**) for the mapping of each signal remain constant at each time step. The previously mentioned backpropagation algorithm can also be applied to RNNs, but using the procedure is problematic because backpropagated gradients tend to either shrink or grow at every step of time. Therefore, the information from long way past might either vanish or explode due to backpropagation. In order to tackle this issue a long short-term memory (LSTM) network can be augmented. An LSTM network is built with an explicit memory to store information for a longer period. (LeCun et al. 2015)

The long short-term network introduced by Hochreiter and Schmidhuber (1997) was designed to get control of the error backflow problems. LSTM network consists of LSTM units which typically include a memory cell, an input gate, an output gate and a forget gate. The inner state of an LSTM unit depends on all of the three gates. The internal state of the LSTM unit, namely the state of memory cell, for the current timestep t can be calculated by (Gers et al. 2000):

$$s_c(t) = s_c(t-1) \circ y_F(t) + g\big(c(t)\big) \circ y_{in}(t), \tag{13}$$

where $\mathbf{y}_\mathrm{F}$ is the forget gate activation, $\mathbf{y}_\mathrm{in}$ is the input gate activation, $\mathbf{c}$ is the cell input signal in which the fed-back output from the previous time step ($\mathbf{y}_{t-1}$) and input of the

current time step ($\mathbf{x_t}$) are combined. The cell input is activated with a hyperbolic tangent function g. Symbol ∘ refers to Hadamard (elementwise) product of matrices. The output of the LSTM unit is defined then as (Gers et al. 2000):

$$y(t) = f\big(s_c(t)\big) \circ y_{out}(t)\,, \tag{14}$$

where f is a hyperbolic tangent function and $\mathbf{y}_{\text{out}}$ is the output gate activation. Figure 13 clarifies the structure of an LSTM unit.

Both the cell state and the cell output at time t are fed back to the unit to calculate the state and output at time $t + 1$, and so on. Each signal connected to forget, input and output gates have weights. These weighted signals are squished through a sigmoid function to scale the activation between 0 and 1. The value of the activation defines if the corresponding gate is "open" or not, 0 meaning the gate is closed and 1 meaning the gate is fully open. As simplified, the meaning of the gates is to decide if information in the internal state should be forgotten (forget gate), what kind of information should be added/updated to the internal state (input gate), and what kind of information of the internal state should be passed to the output of the LSTM unit (output gate). (Gers et al. 2000)

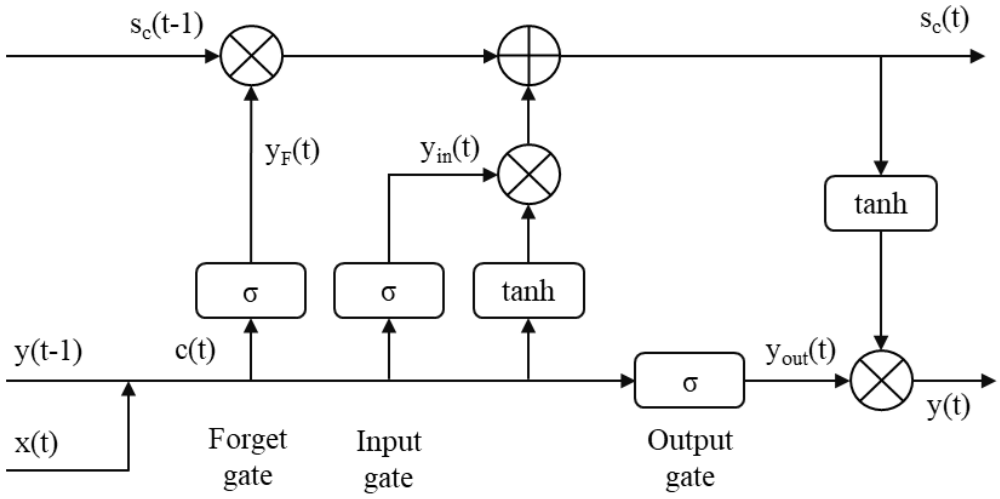

Figure 13. Structure of an LSTM unit. Symbols: ⊗ refers to a Hadamard product and ⊕ to a sum of signals. (retelling Gers et al. 2000).

5.2.4 Ensemble learning methods

Ensemble methods can be useful in regression and classification problems by combining multiple models to ensure better overall performance. There is number of categories of ensemble methods, for example conventional methods which include boosting and random forest, and ensemble methods based on deep learning. In addition to conventional methods, also decomposition ensemble methods have been popular within time-series data. Seasonal time-series can be decomposed by seasonal decomposition to identify patterns in daily, monthly, or yearly interval. Hierarchical ensemble methods are a sub-category of decomposition ensemble methods. In hierarchical methods the time-series data is decomposed hierarchically so that first a predictor is applied to the time-series and then the residue from the first predictor is handled with another predictor. For example, combining a statistical regression, such as ARIMA, with a computational regression algorithm, such as neural networks or support vector regression (SVR), can be useful in time-series forecasting. (Ren et al. 2016)

In ensemble averaging the model output is the average of a collection of predictions. The model structures share a common input, and the outputs are combined to produce the overall output. According to Haykin (1999), if the ensemble averaging approach was replaced with a single complex model structure, the number of parameters needed to be adjusted would be really large. Therefore, the time it takes to train a complex model is likely longer than training a set of simpler models in parallel. Also, the risk for overfitting is presumably lower as the number of adjustable parameters is lower. The idea of ensemble averaging is illustrated in Figure 14. The ensemble averaging is defined as follows:

$$\hat{f}_{bag} = \frac{1}{N}\sum_{n=1}^{N} \hat{f}_n(x), \tag{15}$$

where $\hat{f}_n$ is an original single estimate of model n and N is the total number of original estimates.

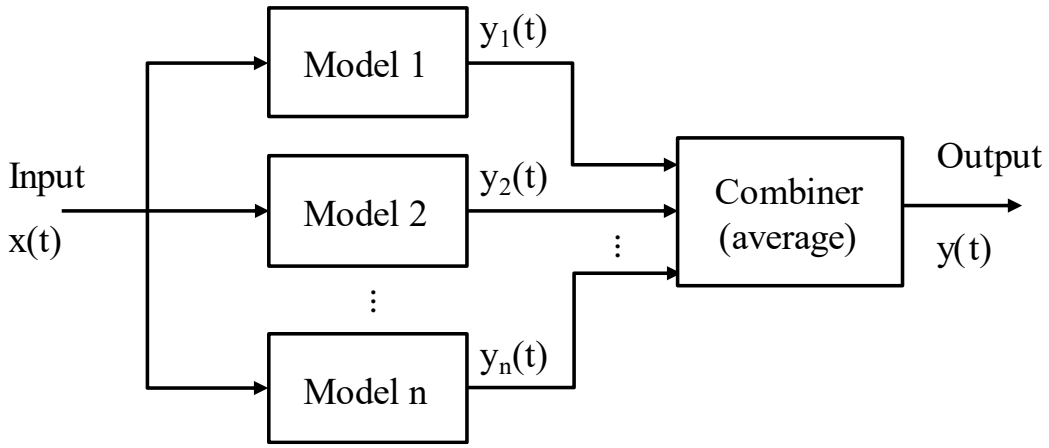

Figure 14. Ensemble averaging (retelling Haykin 1999).

5.3 Variable selection

Variable selection is a crucial part of data-driven modelling problems, and identification of an energy baseline is no exception. There are multiple different approaches to variable selection in the area of energy baseline identification.

The ISO 50006 standard for energy performance indicators refers to variable selection as defining and quantifying relevant variables, especially if the energy performance indicator (EnPI) is model based. The ISO 50001 project shall challenge the organization to understand what kind of relationships lay between the process variables and the energy consumption. The variable selection guided by ISO 50001 and ISO 50006 standards mostly leans on the expertise related to the process of interest. Organization should start by understanding the long-term behavior of the studied energy consumption, for example by simply plotting it as a trend versus time. Only based on this procedure organization may already find evidence of potential seasonality in the energy consumption, or some correlations that the level of energy consumption changes similarly than the level(s) some other relevant variable(s). This usually then leads to deeper study of the relationships, for example by plotting XY-scatters and calculating different correlation metrics. (SFS-ISO 50006 2015)

The variables affecting energy consumption need to be identified case by case, but literature can also guide on that. When energy baseline is identified to a smaller production line of a factory or even for a subprocess, very process-specific variables, such as rotational speed of pumps, might be the most useful ones. Thus, the domain expertise is a great resource to start with. To deal with possible seasonality in the time-series energy consumption data, variables based on calendar effect can be applicable. For example, Elamin and Fukushige (2018) generated multiple dummy variables to consider the seasonality in annual, weekly, and daily levels. With an analysis on the load pattern, they were able to identify similarities in some of the weekdays resulting, for example Monday and Friday being adjacent to weekend, they tend to have similar load profile. In addition to main effects of temperature or seasonal variables, the study also considered the cross effects of those two. With something as complex as energy consumption, a study of the underlying relationships as in-depth as possible is essential, underlining the need for careful exploratory data analysis.

In order to catch the multi-seasonality of the energy consumption, Fourier transformations can be applicable. For example, data can have both a weekly seasonality of 168 hours and a daily seasonality of 24 hours. Fourier transformations can be calculated as follow:

$$x = f\left(\frac{2\pi N}{p}\right), \tag{16}$$

where f is sine or cosine function, N is the length of the dataset, and p is the modeled seasonality (24 or 168).

To use a continuous linear variable such as month number (originally 1–12) or day of week (1–7) in a cyclical form the variables have to be encoded with sine and cosine transformations to cyclical variables. After the encoding, the distance between each value is equal and the continuity is guaranteed. Trigonometric transformations for encoded variables can be calculated as follows:

$$x = f\left(2\pi \frac{x_{original}}{\max(x_{original})}\right), \tag{17}$$

where f is sine or cosine function and $x_{original}$ is the original value of the variable.

If domain expertise is not available and the most relevant variables are not known, variable and feature selection methods can be convenient for this purpose. It can also lead to deeper knowledge of the prevailing relationships. With today's measurement technology it is usual that lots of data is gathered real-time from the processes, resulting in datasets that may include thousands of variables available. Guyon and Elisseeff (2003) described the three objectives of variable selection as follows: it improves the predictors' performance in terms of accuracy, it provides faster and less computationally expensive predictors, and it provides a deeper understanding of the process from which the data is gathered.

The earlier mentioned correlation-based criterion is a practical starting point for variable ranking but limits to linear dependencies, and therefore nonlinear preprocessing might be needed. Variable ranking can also be approached using information theoretic ranking criteria, such as mutual information between each variable and the target. With variable ranking it is also computationally more cost efficient to continue with various variable selection methods such as forward selection algorithm. Forward selection algorithm is a method in which the input variables are set to the model one by one, for example in an order based on the results of the selected variable ranking method. After a variable has been added to the model, model is evaluated, and procedure is continued until adding a new variable does not improve model accuracy anymore. Forward selection can lead to a situation where a pair or a group of variables which are useless independently but useful together might not get tested together. In forward selection they would get tested together only if the other variables were added to the model earlier. Therefore, backward elimination is more suitable algorithm in some cases. Backward elimination starts with a model with all the studied variables and proceeds by eliminating variables one by one until model performance does not improve anymore. As an advantage for backward elimination is the possibility to overcome the previously mentioned problem

with a pair or a group of variables which are useful together. On the other hand, if in the end the optimal set of variables is much smaller than the dimension of the data, lots of models with useless variables get evaluated during the process. Therefore, backward elimination can be computationally more expensive. Backward elimination and forward selection methods are so called deterministic wrapper selection methods, and variable ranking is a filter method. There are also many other variable selection methods available and the decision for the used method can be based on for example the complexity of the problem and on the computational resources in use. (Guyon & Elisseeff 2003)

From energy baseline perspective, the importance of understanding and compiling large and statistically representative datasets has been recognized for example by Granderson and Price (2014). On their study related to energy baselines of buildings the dataset was narrow in terms of the number of variables, consisting only weather data and electric demand data. They stated that from a large representative dataset the types of different buildings could be identified according to their predictability in respect of energy consumption. This would then lead to more accurate and reliable quantification energy consumption and further on selecting ideal candidates for energy savings measures.

5.4 Model validation and adaptation

Model validation is an integral part of model identification to ensure that the identified model generalizes also to data independent from the training data. The way how the data is partitioned to training and test datasets depends on the problem and available data. With time-series data, especially if it is gathered on a long continuous period, it might be practical to just hold out a part of data for model testing purposes. Though, it must be taken care that the used test data is representative enough in the big picture. Cross-validation techniques, in which data points are used in both training and testing, can also be used in time-series problems but the assumptions of independent and identically distributed data do not hold anymore (Guyon & Elisseeff 2003).

Hyndman and Koehler (2006) categorized time-series forecasting accuracy measures in four groups: scale-dependent measures, measures based on percentage errors, measures based on relative errors, and relative measures. From scale-dependent measures Mean Square Error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are often used. A scale-independent Mean Absolute Percentage Error (MAPE) is also frequently used to measure the performance of a time-series forecast. MAE and MAPE are defined as follows:

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|y_t - \widehat{y_t}|, \tag{18}$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\frac{|y_t - \widehat{y_t}|}{y_t} * 100\ \%, \tag{19}$$

where N is number of datapoints, y_t is measured value at time t and $\widehat{y_t}$ is the corresponding forecasted value.

Adaptation is important to ensure the validity of a model in long-term on-line use. As the operating conditions might change, the original set of parameters identified during the first training might not represent the current situation anymore, thus raising the model error. This shifted data does not work with the underlying assumptions, and therefore the model needs to be retrained and readjusted. Bakirov et al. (2021) defines adaptation as “changes in model training set, parameters, and structure”. There are multiple strategies for automated adaptation.

According to Bakirov et al. (2021) it can be beneficial to incorporate the most recent data in the training dataset. On the other hand, it is important to ensure the validity of the data before using it in model training. Measurement validation together with model validation gives information if the forecast accuracy was bad because the model parameters need an update or because the test data did not represent the training data, and thus the forecast accuracy for that period looked bad against the test data.

Measurement validation methods in terms of distribution similarity measures were described in Chapter 3.2.1.

6 MATERIALS AND METHODS

The experiments with the data were executed in MATLAB® environment using various toolboxes such as Deep Learning Toolbox (MathWorks 2022a), Statistics and Machine Learning Toolbox (MathWorks 2022b), and Econometrics Toolbox (MathWorks 2022c).

6.1 Data sets

The data used in the experiments was imported from a time-series database, ABB Ability™ History, in which the continuous measurements coming from the automation systems are aggregated to different history levels such as hourly sums, averages and minimum or maximum values. In this study the hourly and daily averages of the variables were used. Therefore, a normal year (365 days/year) consists of 8760 1-hour-average data points per variable. The datasets consist of energy consumption data from two factories, A and B, from two years, 2020 and 2021. Year 2020 was a leap year, and therefore it consists of 366 days on that year. Thus, the total number of data points for each factory is 17544. Change in the timestamp due to daylight saving time (summertime) was visible on the data and it was considered in data preprocessing phase. For the results presented in the thesis, the energy consumption values were scaled as a percentage between 0–100% of maximum energy consumption. The hourly average profiles of the energy consumptions of factories A and B in 2021 are presented in Figure 15.

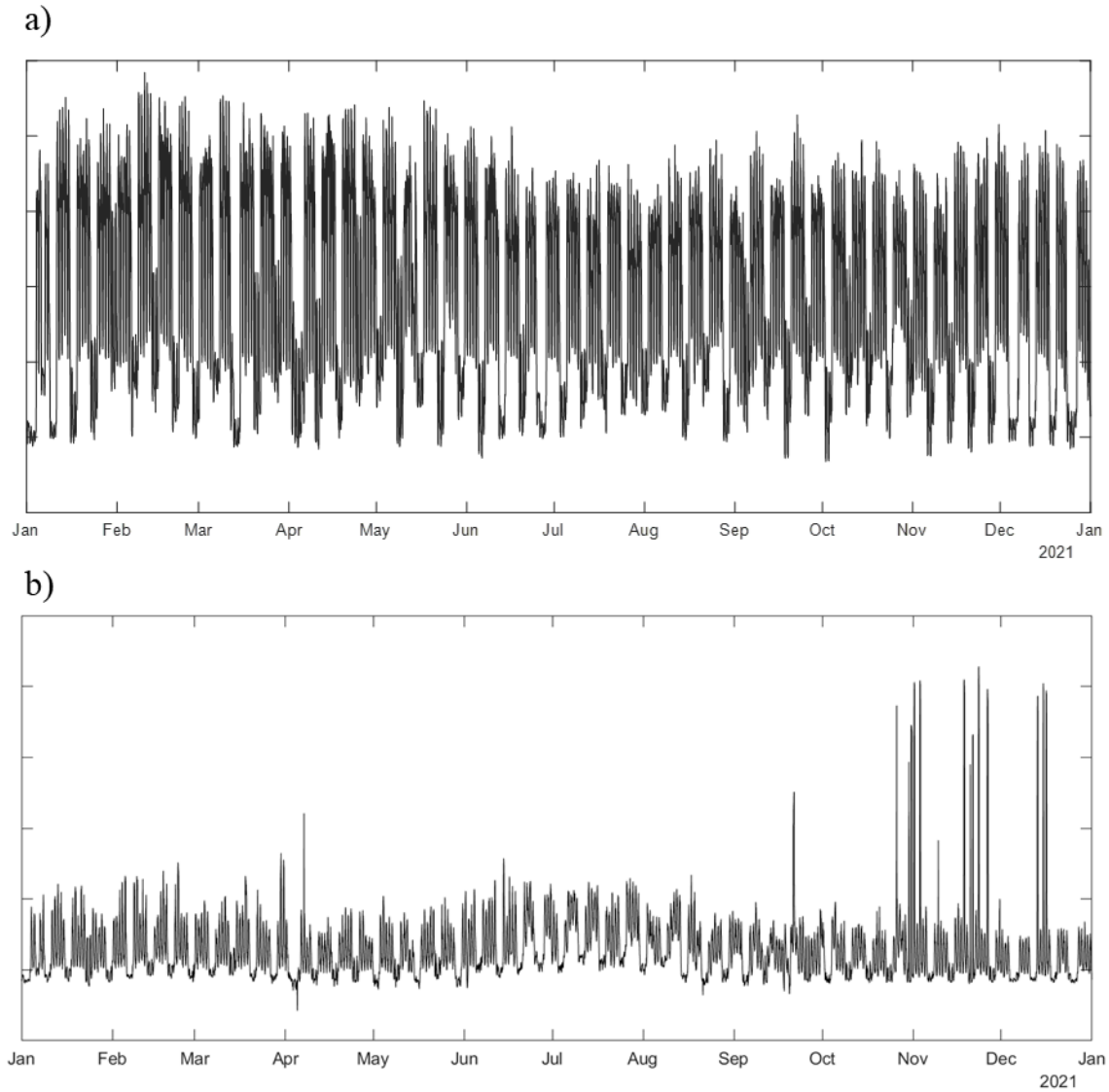

Figure 15. Hourly average energy consumption profiles of factories A (a) and B (b) in 2021.

6.2 Simulation scenario

In this study the on-line scenarios are simulated week-by-week so that the upcoming week is forecasted after which the forecast accuracy is validated against the measured values. In real life these measurements would be available after the forecasted week has passed. As the studied time-series forecasting methods are supposed to operate on-line, model adaptation process is an integral part of the whole. This study proposes an adaptation procedure which consists of measurement validation, model validation, and a possible model parameter estimation.

Because the used data was not labeled in terms of normal or abnormal operating conditions, an assumption had to be made that the specific consumption of the process remains constant throughout the studied period. Therefore, the forecast error, namely the error between the baseline forecast and the observer consumption, should be zero for a good model.

The scenario starts with two weeks of pre-collected data: first of those two is used as training data and later as test data. The second week is forecasted with a model trained with the data of the first week and forecast accuracy is validated against the data of the second week. If the model accuracy is not good enough, the test data is joined to the training data and model is trained with the new data. Then a forecast is made for the upcoming third week. After that, measurements are collected for a week, and procedure is started again with three weeks of data. The week-ahead forecast is considered as the energy baseline for the upcoming week. When the energy consumption of week three has been observed, the measurements are validated against the data from the previous weeks with metrics describing similarities of two observation populations, such as Kullback-Leibler divergence (Equation 2), Jaccard index (Equation 4), and histogram intersection (Equation 5).

The combination of training and test data is set to be a rolling window of maximum 52 weeks. This means that as the on-line use of the model starts, the train/test dataset is grown until it has data for 52 consecutive weeks. Before 53rd week is added, the first

week of the data set is dropped out. The reason for the 52-week rolling window is that the training/test dataset should capture the long-term seasonal patterns on a year-level as well as the national holidays and other anomalous operating periods. The proper dataset to be selected for forecasting purposes must be considered case-by-case. For example, big changes in the process or measurements might change the energy consumption after the change, and therefore the upcoming data might not match the distribution of the pre-change data.

In this study the initial data was set to be 52 weeks meaning that model performance in week-ahead forecasting was evaluated in a scenario where the rolling window is already "full". Thus, the simulation starts after one year of data is used for training and as the simulation proceeds, the oldest data is dropped out. This was done to guarantee impartiality between the experiments with different model structures. The studied machine learning and deep learning methods need large amounts of training data. Hence, the earlier described scenario with only two weeks of initial data would probably favor more simpler methods early in simulation.

6.3 Exploratory data analysis

6.3.1 Case 1: Factory A

The energy consumption of factory A follows a noticeable pattern in which peaks of energy consumption can be seen during daytime from Monday to Friday. The functions of the factory can be roughly separated to energy-intensive component manufacturing lines and to less energy-intensive assembly lines. Component manufacturing processes operate in two shifts daytime from Monday to Friday whereas assembly lines operate around the clock from Monday to Friday. Operations during weekend are irregular and decided during the week based on the demand of the goods, and on the available human resources. Because more accurate production calendar was not available for modelling purposes, weekdays were labelled as operating days and weekends as non-operating days. In addition, the national holidays in Finland were labeled as non-operating days, as there was a clear sign that energy consumption drops down to a level of non-

operating day during the national holidays. Histogram of the total electricity consumption in 2021 can be seen on Figure 16. The multimodal shape of the histogram indicates that there are multiple operation points in terms of energy consumption.

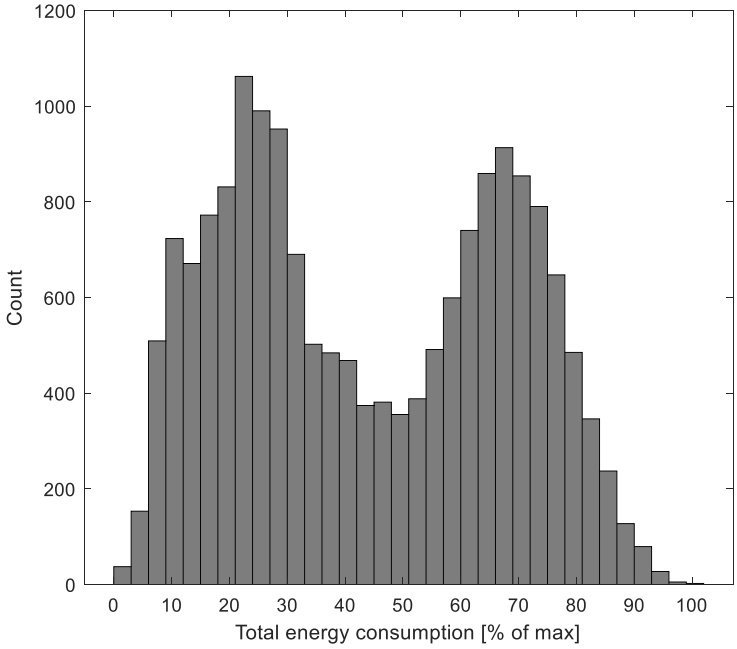

Figure 16. Histogram of energy consumption of Factory A in 2021.

While taking a closer look on the energy consumption, the daily and weekly seasonality is clearly visible. A heat map of typical week is presented on Figure 17. As seen on the heatmap, the energy consumption is distinctly higher daytime from Monday to Friday. During nighttime from Monday to Friday as well as daytime on weekends the energy consumption is about half the consumption of daytime of production days. Otherwise, the consumption is much lower, being lowest on weekend nights.

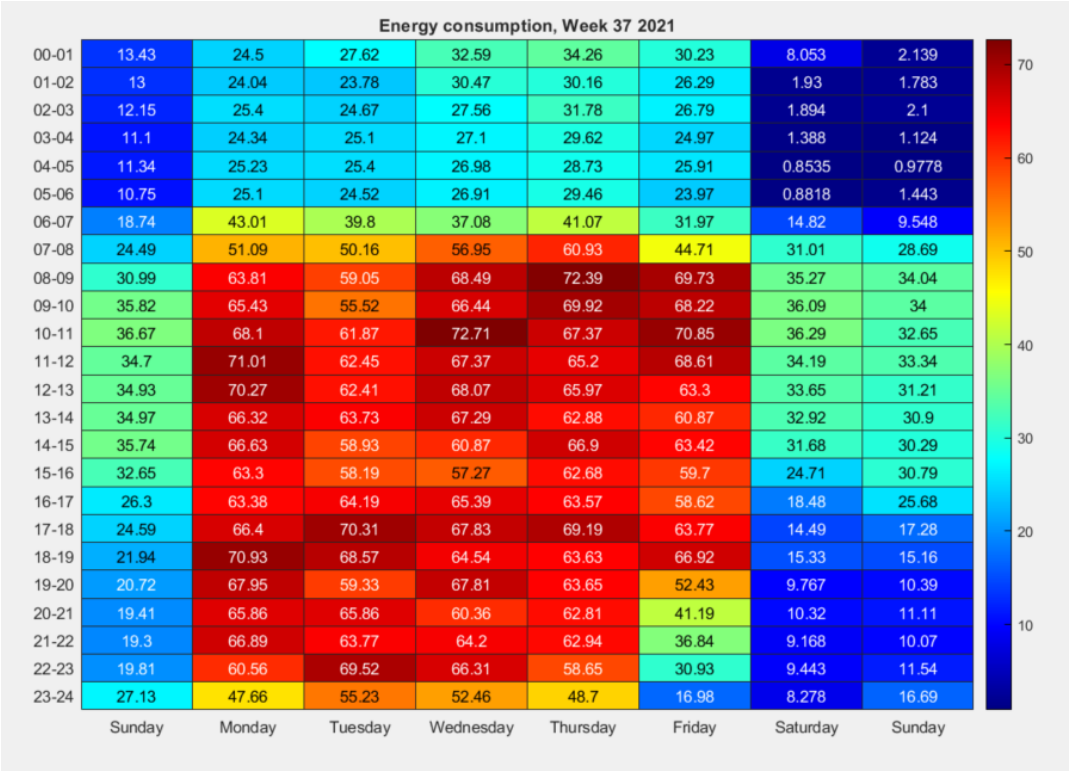

Figure 17. Heatmap of the energy consumption on a typical operating week [% of max].

The seasonality of 24 hours and 168 hours, latter referring to weekly seasonality, can be seen on the autocorrelation plot of the energy consumption. With maximum lags of 200 hours, there are clear peaks in the autocorrelation with lags of 24 hours and 168 hours. The autocorrelation plot can be seen on Figure 18. The knowledge of underlying seasonality is crucial when selecting a suitable model structure and variables in the model identification process.

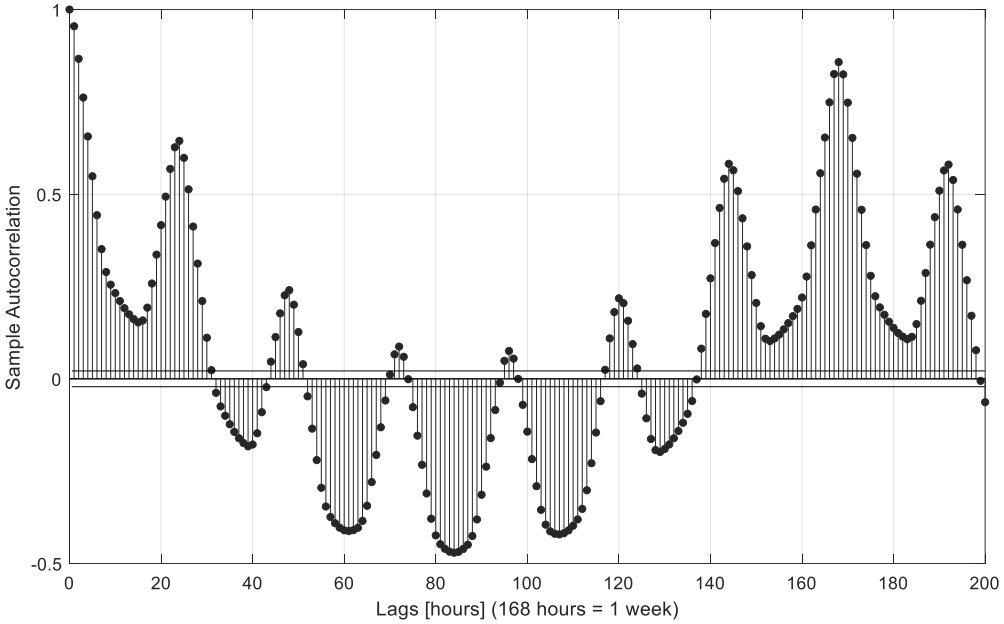

Figure 18. Autocorrelation plot of energy consumption in 2021 with maximum lags of 200 hours.

Due to the lack of any production schedule, individual days could not be labeled, and it caused difficulties especially on weekends. Figure 19 illustrates a three-week period of energy consumption of factory A in June 2021. The lower consumption periods are weekends and higher consumption periods are weekdays. Also, June 25th, 2021, was a national holiday, and it clearly has a similar consumption profile than the following regular weekend days. The difficulty with modelling the weekends is on the varying energy consumption level. As stated earlier, the irregular character whether the factory is operating on a certain weekend or not, cannot be modelled without a clear labeling for each day based on the production schedule. On Figure 19 the weekend period in the middle (June 19th–20th) is clearly distinct from the other two weekends based on energy consumption level. However, without clear information it is difficult to decide whether the days should be considered as operating days or not.

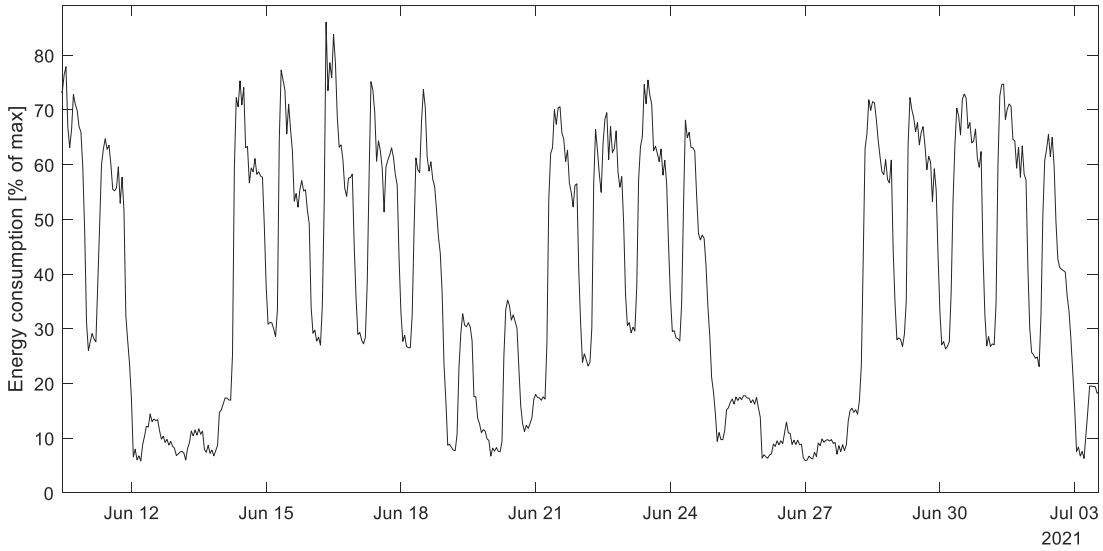

Figure 19. Typical hourly average profile for factory A.

6.3.2 Case 2: Factory B

The generalization ability of the selected forecast modelling technique was evaluated with data of factory B which is also in the field of discrete industry. In addition, three other variables were available for factory B: total district heating consumption, total water consumption, and outside temperature. These variables were used to study could the forecast accuracy improve with them utilized as predictor variables.

Factory B shares similar energy consumption pattern than factory A: energy-intensive periods appear daytime on weekdays whereas nighttime and weekends are periods with less energy consumption. The energy consumption data of factory B from last two months of 2021 had multiple unexplainable high values which were presumed as anomalies. The magnitude of these anomalies was twice the value of the maximum of rest of the data. For this study the data from November and December 2021 was left out of the used dataset because of the unexplained anomalies. Histogram of the whole data of 2021 is in Figure 20.

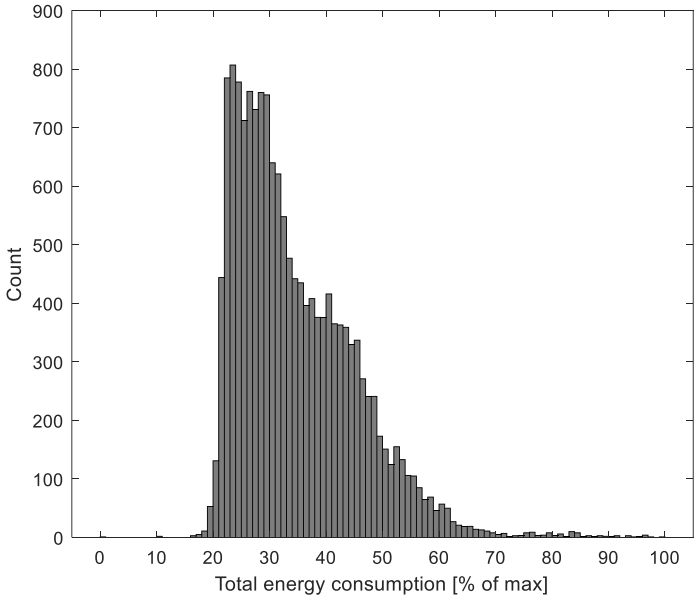

Figure 20. Histogram of energy consumption of factory B.

The daily and weekly periodicity of the energy consumption of factory B can be verified with an autocorrelation plot which is shown in Figure 21. The autocorrelation plot appears to have quite similar profile than the autocorrelation plot of factory A shown in Figure 18.

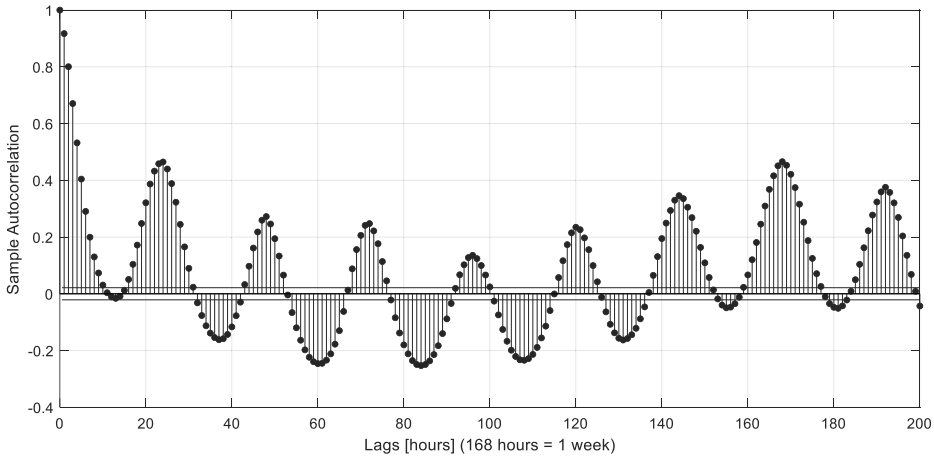

Figure 21. Autocorrelation plot of energy consumption in 2021 with maximum lags of 200 hours.

6.4 Predictor variables

Used predictor variables were derived to interpret the calendar effect. Predictor variables were used in all model structures. With only limited subject matter expertise available, the variables were designed only based on carefully executed exploratory data analysis. A list of the selected predictor variables for modelling is presented in Table 1.

Table 1. Selected predictor variables.

Variable	Explanation	Value
x_1	168 h sine Fourier-term	[-1,1]
x_2	168 h cosine Fourier-term	[-1,1]
x_3	24 h sine Fourier-term	[-1,1]
x_4	24 h cosine Fourier-term	[-1,1]
x_5	Encoded cyclical month (sine)	[-1,1]
x_6	Encoded cyclical month (cosine)	[-1,1]
x_7	Encoded cyclical day of week (sine)	[-1,1]
x_8	Encoded cyclical day of week (cosine)	[-1,1]
x_9	National holiday (0) or not (1)	0 or 1
x_{10}	Weekday (0) or weekend (1)	0 or 1
x_{11}	Weekday - time 06:00-22:00 (1), else (0)	0 or 1
x_{12}	Weekday - time 22:00-06:00 (1), else (0)	0 or 1
x_{13}	Weekend/national holiday - time 06:00-22:00 (1), else (0)	0 or 1

Variables x_1–x_4 were designed in order to catch the multi-seasonality of the energy consumption. The Fourier transformation was calculated as presented in Equation (16). The month number (1–12) and day of the week (1–7) were encoded using Equation (17) resulting in variables x_5–x_8. National holidays on weekdays for years 2020 and 2021 in Finland were labeled in variable x_9, and the national holidays are presented in Table 2.

Table 2. National holidays on weekdays in Finland (2020 and 2021).

Date (dd.mm.yy)	National holiday
1.1.2020	New Year's Day
6.1.2020	Epiphany
10.4.2020	Good Friday
13.4.2020	Easter Monday
1.5.2020	May Day
21.5.2020	Ascension Day
19.6.2020	Midsummer's Eve
24.12.2020	Christmas Eve
25.12.2020	Christmas Day
31.12.2020	New Year's Eve
1.1.2021	New Year's Day
6.1.2021	Epiphany
2.4.2021	Good Friday
5.4.2021	Easter Monday
13.5.2021	May Day
25.6.2021	Midsummer's Eve
6.12.2021	Independence Day
24.12.2021	Christmas Eve

6.5 Model structures for week-ahead forecasting

For factory A five different model structures were used: multivariable linear regression (MLR), autoregressive moving average model with exogenous variables (ARMAX), non-linear autoregressive model with exogenous variables (NARX), and long short-term memory network (LSTM). In addition, an averaging ensemble model combining

seasonal autoregressive moving average model (SARMA), NARX, and LSTM was used.

6.5.1 Multivariable linear regression

MLR model was trained with hyperparameter optimization algorithm which selected the optimal model for each training by evaluating support vector machine (SVM) and least squares learners, ridge, and lasso regularizations, and different hyperparameters of those. Used optimizer depends on the optimized structure at each iteration. Optimizers used in the search are for example stochastic gradient descent, asynchronous stochastic gradient descent, and dual stochastic gradient descent.

6.5.2 ARMAX

A linear time series model, autoregressive moving average model with exogenous variables (ARMAX), was designed based on the exploratory data analysis phase. The selected lags for autoregressive and moving average components were 1 and 2 for both, therefore the model had a form ARMAX(2,0,2). Model constant (β_0) was set to be 1 due to instability with the present exogenous regression components if constant was other than 1. Thus, the ARMAX-model had following form:

$$y_t = c + \sum_{i=1}^{k} \beta_i X_{i,t} + \sum_{p=1}^{2} \phi_p y_{t-i} + \sum_{q=1}^{2} \theta_q \varepsilon_{t-1} + \varepsilon_t \,, \tag{20}$$

where y_t is the observation at time t, ε_t is random error at time t, c is the model constant, β_i is a vector of regression coefficient parameters of exogenous variables, $X_{i,t}$ is a vector of observations of exogenous variables at time t, ϕ_p is a coefficient of AR operator of order p, and θ_q is a coefficient of MA operator of order q. The model coefficients β_i, ϕ_p, and θ_q were solved with an optimizer based on sequential quadratic programming algorithm.

6.5.3 NARX

In NARX-model the non-linear function was a neural network which was trained in open-loop and closed after the training in order to make the multistep-predictions. NARX-network consisted of 10 hidden neurons and the inputs were delayed by two timesteps resulting in AR(2) model. In closed-loop form of NARX-network the predicted output is fed back to the input layer and delayed with autoregressive lag. In addition, the exogenous variables are delayed with same autoregressive lag. The NARX-network was trained with Bayesian regularization. The open and closed loop structures of the NARX-network are illustrated in Figure 22.

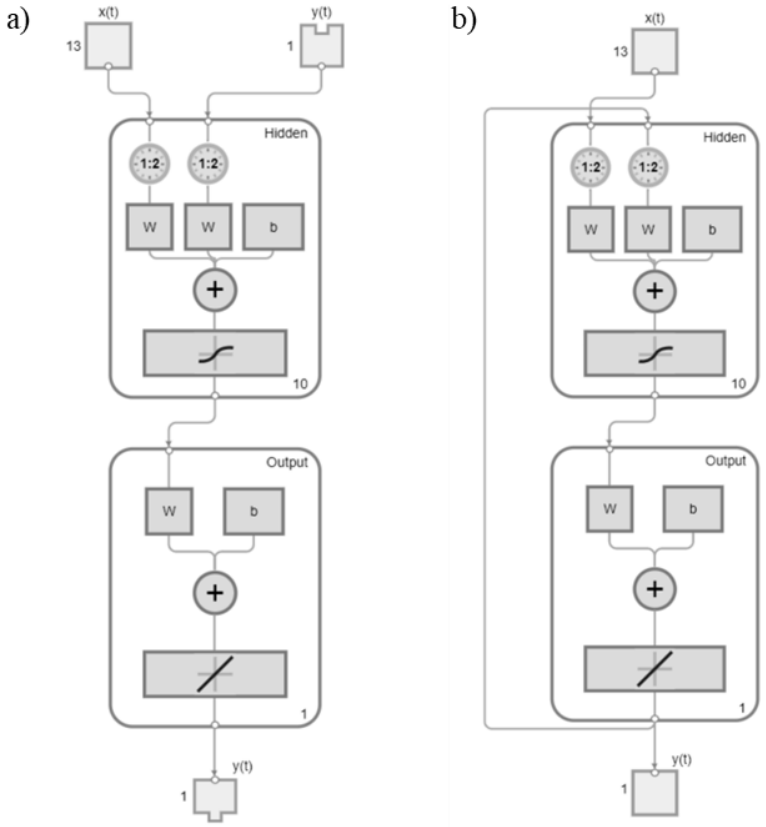

Figure 22. a) Open-loop and b) closed-loop structures of the NARX-network.

In the training of the NARX-network 85% of the data was used to train the network and the rest 15% of the data was used for internal validation. The data was divided randomly. The activation function on the neurons in the hidden layer was hyperbolic tangent function which scales the output of a neuron between -1 and 1. The transfer function of the output layer was a pure linear transfer function.

6.5.4 Long short-term memory

The LSTM network in this study consisted of four layers: sequential input layer, LSTM-layer, fully connected layer, and regression layer. The output of the LSTM-layer was sequential, and the LSTM-layer had 50 hidden units. The optimization was done using Adam-algorithm which is a stochastic gradient-based optimization method (Kingma & Ba, 2014). Simplified structure of the LSTM-network is illustrated in Figure 23.

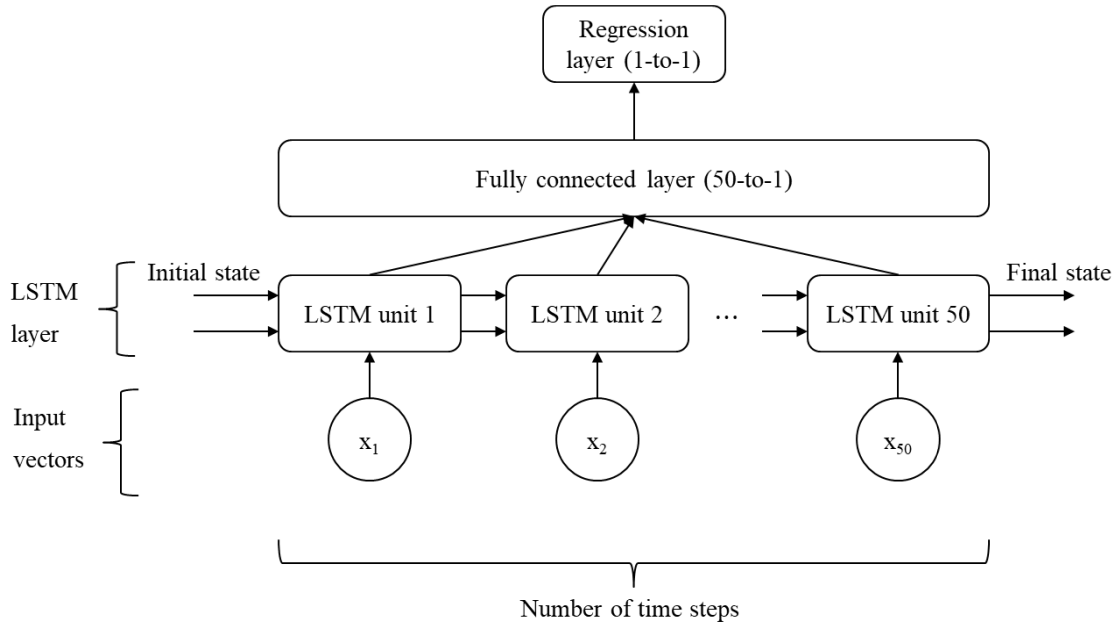

Figure 23. Simplified structure of the LSTM-network.

6.5.5 Ensemble-model

Lastly, an ensemble model of SARMA-model, NARX-model and LSTM-model was tested for factory A. The final forecast of the ensemble model was calculated as an arithmetic mean of the individual forecast of each component.

The SARMA-model had a structure of SARMA(0,0,0)$(2,0,2)_{168}$ indicating 168-hour seasonal autoregressive and moving average lags of two were considered in the model. NARX and LSTM models were the same as described in Chapters 6.5.3 and 6.5.4.

6.6 Time-series anomaly detection

Time-series anomaly detection for the used hourly average energy consumption dataset was studied with statistical process control, isolation forest, and autoencoder approaches. Anomaly detection methodologies were studied with the energy consumption data of factory A so that the used data was filtered by selecting only data of weekdays (excluding national holidays) between hours 08–23. The data was selected so in order to get a normally distributed dataset including data from the most energy-intensive daytime period. Histogram of the selected dataset is on Figure 24.

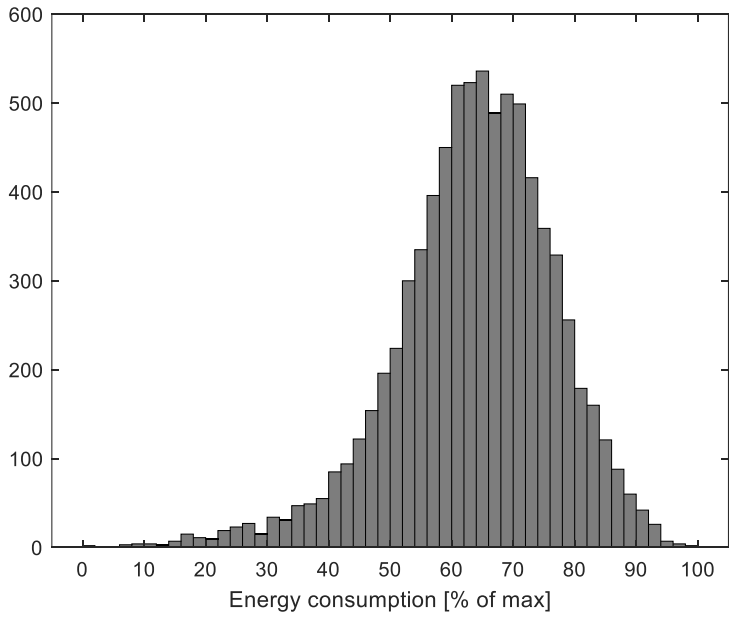

Figure 24. Histogram of the selected dataset.

6.6.1 Statistical process control

The theoretical background of statistical process control and how it can be established in time-series anomaly detection was shortly covered in Chapter 4.1. For testing the statistical process control approach on energy consumption daily averages were calculated for the selected dataset. This was done with an idea to capture longer term deviations than hourly average data points. An appropriate measurement resolution for statistical process control with energy consumption data was considered to be daily averages, shift averages, or even weekly averages, suggested by interviewed subject matter experts. Because the data was totally unlabeled anomaly-wise, an assumption had to be made that the data did not have any anomalies in its raw form. To test the anomaly detection capabilities, few point anomalies and positive bias shift periods were added to the daily average data in random positions. The following Western Electric (1956) and Nelson (1984 and 1985) rules were selected for the experiments in this study:

- **Nelson 1:** One measurement is beyond Zone A
- **Nelson 2:** Nine (9) or more consecutive measurements on the same side of the population mean
- **Nelson 3:** Six (6) or more consecutive measurements are continually increasing or decreasing
- **Western Electric 2:** Two out of three (2/3) consecutive measurements are beyond Zone B
- **Western Electric 3:** Four out of five (4/5) consecutive measurements are beyond Zone A

6.6.2 Isolation forest

The interest regarding isolation forest algorithm on this study is on point-anomalies that may occur on energy consumption data on hourly and daily average levels. The anomaly score of each observation (x) was calculated as:

$$Score(x) = 2^{\frac{-E|h(x)|}{c(n)}}, \quad (21)$$

where $E|h(x)|$ is the average path length over all isolation trees in the forest, and $c(n)$ is the average path length of failed searches in a binary search tree of n observations. The closer the anomaly score is to value 1 the more anomalistic it is considered to be. Anomaly scores closer to 0 are considered as normal points. Contamination fraction is an important parameter in the training phase of an isolation forest. With contamination fraction is defined the percentage of anomalies in the training data. Contamination fraction does not affect the model itself but has an effect on how the deciding threshold is set.

For the experiment the hourly average energy consumption data was contaminated with point anomalies. Point anomalies were added so that they occurred randomly with a random probability of 0.1–0.4%. These anomalies represent situations where there is faultiness in a measurement device, or some other error which causes a spike in the data. The contamination fraction was set randomly within 0.1–1%, thus not matching the actual anomaly frequency.

6.6.3 Autoencoder

Autoencoder was trained with 50 neurons in the hidden layer. The activation function for encoder was a saturating linear transfer function, and the activation function for decoder was a pure linear transfer function. L_2-regularization was applied to the cost function with a weight 0.001. Also, sparsity regularization was applied. Sparsity regularization is attempting to impose a constraint on the sparsity of the output. The cost function for training was a mean squared error function adjusted for a sparse autoencoder. The autoencoder was trained with scaled conjugate gradient descent algorithm.

Autoencoder was tested with original raw data of factory A so that the hourly average energy consumption data of year 2020 was used for training, and the data of year 2021 was reconstructed through the autoencoder. The anomaly detection was then based on

the reconstruction error. An anomaly score was calculated for each reconstructed data point as follows:

$$AS_i = \frac{\left\| y^i_{test} - \hat{y}^i{}_{test} \right\|_2 - \min\left(\left\| y^i_{train} - \hat{y}^i{}_{train} \right\|_2 \right)}{\max\left(\left\| y^i_{train} - \hat{y}^i{}_{train} \right\|_2 \right)}, \tag{22}$$

where $\|x\|_2$ indicates an L_2-norm of x, subscript $test$ refers to a datapoint which belongs to test data (reconstructed data of 2021), subscript $train$ refers to a datapoint which belongs to training data (data of 2020), y refers to an actual measurement, and $\hat{y}$ refers to a reconstructed datapoint. In other words, the anomaly score is the reconstruction error of a datapoint normalized with respect to the training data. If a datapoint of the test data differs significantly from the training data, the anomaly score can get a value higher than 1. The threshold which decides whether a datapoint is anomalous (1) or not (0) was defined based on three-sigma rule. Mean (μ) and standard deviation (σ) for the anomaly scores were calculated and the threshold was set as follows:

$$Anomaly = \begin{cases} 1 & AS_i \geq \mu + 3\sigma \\ 0 & otherwise \end{cases}. \tag{23}$$

7 RESULTS AND DISCUSSION

The results of energy consumption forecasts for factories A and B as well as time-series anomaly detection studies for factory A are presented in this chapter.

7.1 Week-ahead forecasting

The accuracy of the forecast was evaluated with mean absolute percentage error (MAPE). In addition, the model retraining frequency was monitored for each simulation scenario. Retraining frequency means the percentage of in how many weeks the model was retrained based on the validation result:

$$Retraining\ \% = \frac{Number\ of\ retrainings}{Total\ length\ of\ the\ scenario} * 100\ \%, \tag{24}$$

where the total length of the scenario is 49 weeks in this case. 49 weeks of the data was left over after first 52 weeks of the data set were initialized for model training.

7.1.1 Case 1: Factory A

In terms of forecast accuracy, the averaging ensemble model outperformed other model structures with MAPE of 9.3%. Ensemble model was retrained on 36.7% of the weeks. The forecast accuracy results of all studied model structures are shown on Table 3. The recurrent networks, NARX and LSTM, performed equally well MAPE-wise. Although LSTM-network was retrained more frequently than NARX-network. Also, MLR and ARMAX models had similar performance judged by MAPE. More complex model structures performed better on this week-ahead forecasting scenario than the simpler ones. The retraining percentages sound moderate. For example, the best performing model was retrained on average close to every three weeks. Considering that on this scenario the model should be retrained only once per week maximum, the computational cost of retraining is not growing too big even though the model structure is a complex deep learning network or an ensemble model.

Table 3. Results of week-ahead forecasting simulations

Model structure	MAPE [%]	Retraining [%]
MLR	13.8	63.5
ARMAX	13.4	84.7
NARX	10.5	38.8
LSTM	10.5	49.0
Ensemble	9.3	36.7

Figure 25 illustrates the profiles of the forecasts of studied model structures. The time period is same for each graph: a three-week period in June 2021. The selected period consists also a national holiday, June 25th, 2021.

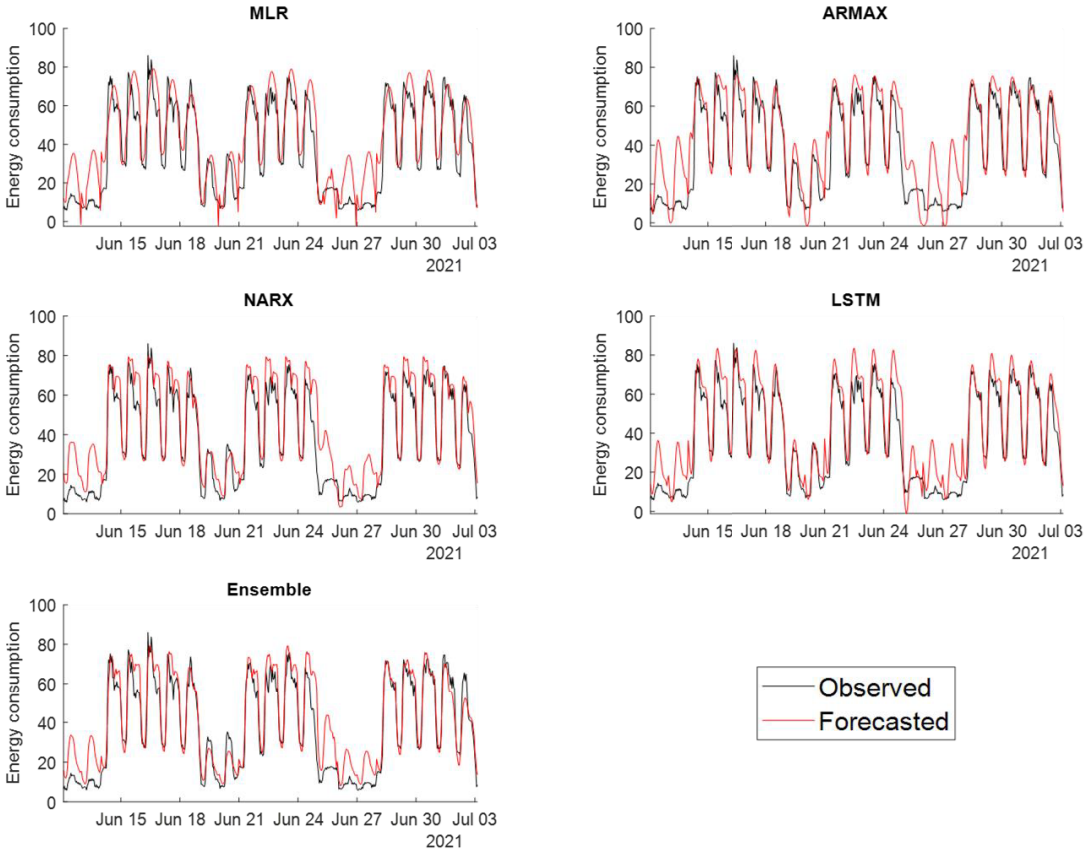

Figure 25. A three-week period of forecasts compared to observed energy consumption [% of max energy consumption].

By looking visually at the shape of each histogram in Figure 26, it seems that every model structure was able to produce forecasts so that the forecast error follows a normal distribution. Also, the average error seems to be close to zero error for each model structure.

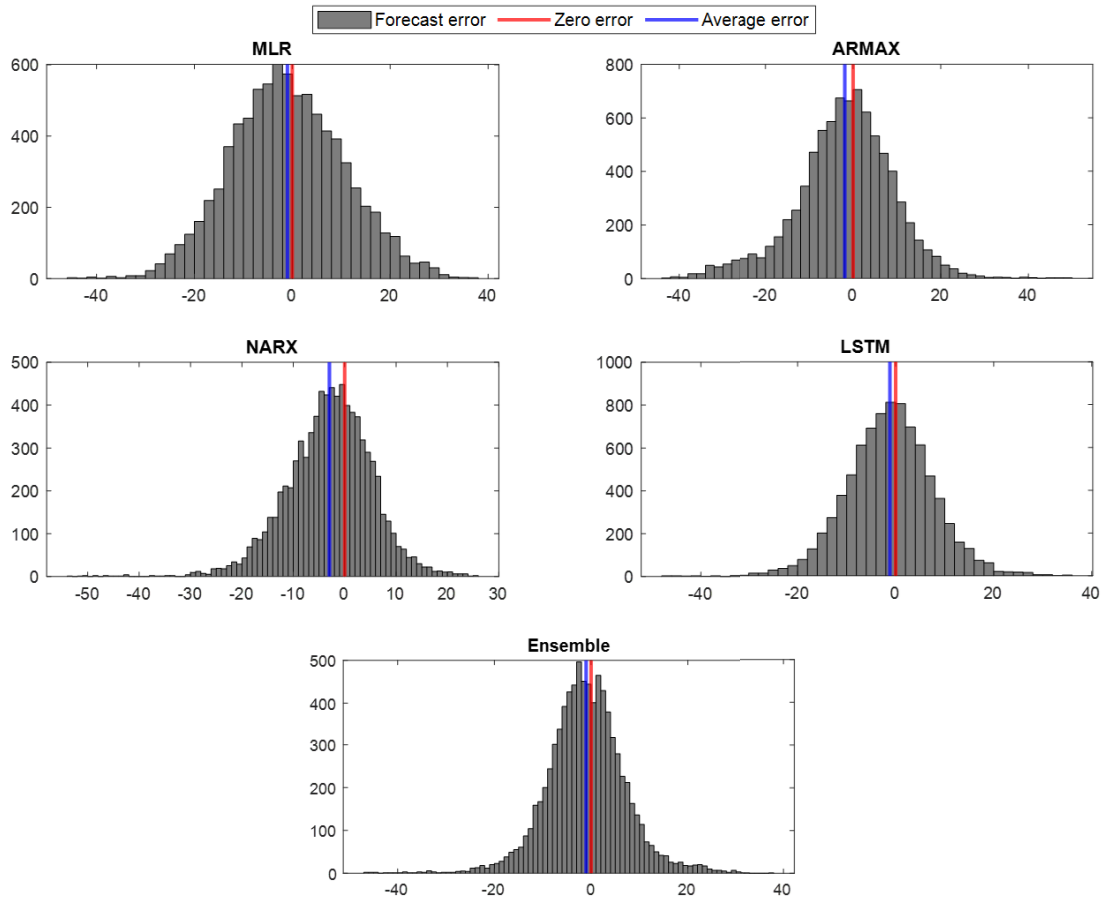

Figure 26. Histograms of forecast errors [% of max consumption].

To dig in deeper on the shape of the error histogram, normal probability plots of the forecast error were drawn (Figure 27). If the errors come from a normal distribution, the black markers shall follow the red line completely. The error produced by multiple linear regression model seems to be almost perfectly normally distributed. The other normal probability plots seem to have little bit of deviation on both tails. This indicates that the tails of the distribution are long, namely there are errors that lay far away from the center. By looking at the forecast profiles presented in Figure 25, these bigger

errors, especially on ARMAX, NARX, LSTM, and ensemble models, are present at weekend days (for example June 25th–26th), which could not be labeled more precisely. The earlier described problem that there are also some weekends with higher energy consumption because of production, causes instability on the forecast precision. While the overall accuracy of the MLR model is less than others', the model forecast does not overestimate and underestimate these weekend periods as extensively as the other models do. With better understanding of the production schedule allowing to label the days better, the forecast could be improved. This underlines the need of domain expertise. Utilizing only history data of the energy consumption in modelling can lead to fairly good forecast results, but the calendar effect could be applied much more efficiently, if the causes and the consequences were known in more detailed manner.

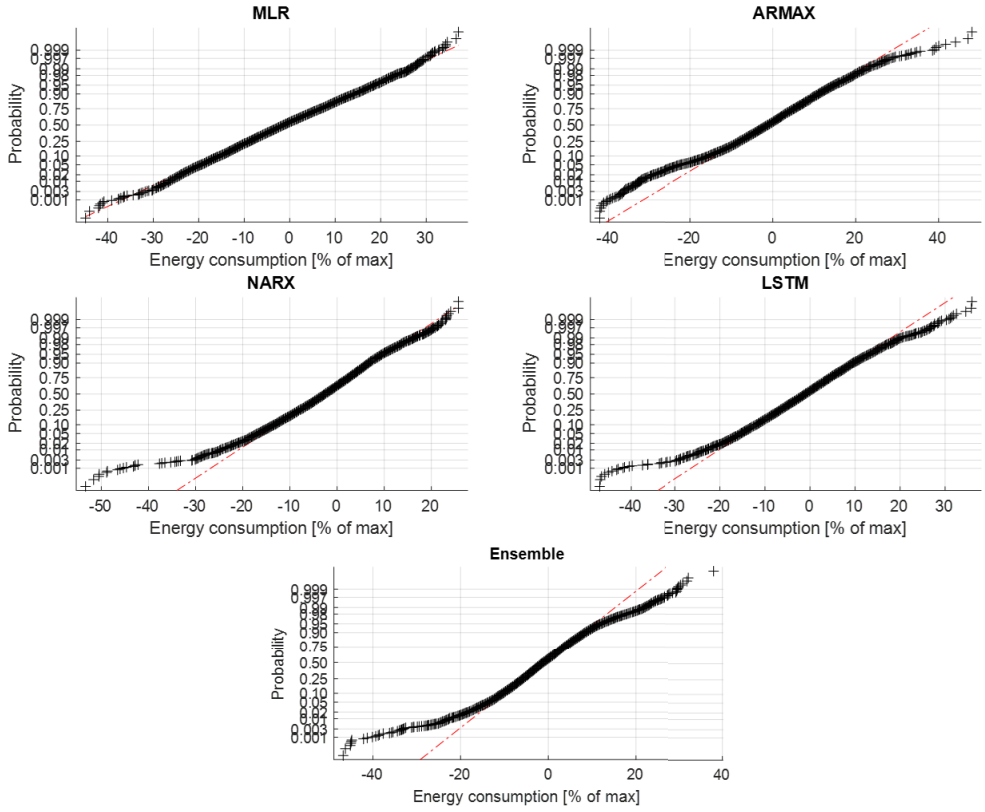

Figure 27. Normal probability plots of forecast errors.

To consider the left-over seasonality in forecast, the residual autocorrelation plots with maximum lag of 200 hours were drawn. Autocorrelation plots are shown on Figure 28. Both 24-hour and 168-hour seasonalities are somewhat visible on every autocorrelation plot. On autocorrelation plot of the ARMAX model it seems that 24-hour seasonality has been covered by the model quite well but a spike at lag 168 hours expresses that there is still a clear weekly seasonality in the forecast error. For NARX, LSTM, and ensemble models the weekly seasonality seems to vanish from the forecast error. Instead, the 24-hour seasonality appears to be visible on the forecast error autocorrelation plots.

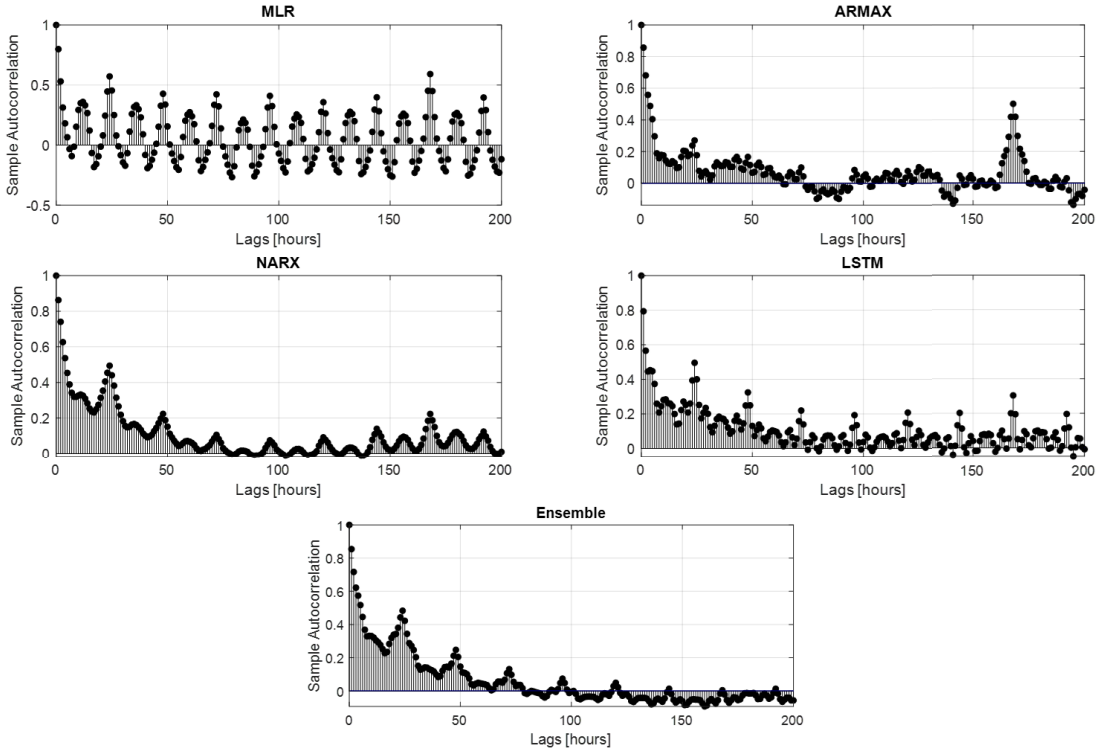

Figure 28. Autocorrelation plots of forecast errors.

Scatter plots of observed values versus forecasted values are presented in Figure 29. The black dashed line indicates a perfect fit ($x = y$), namely the forecasted value is exactly the same than the actual observed value. Most of the points lay around the dashed line while some of the points are further away referring to decreased forecast accuracy.

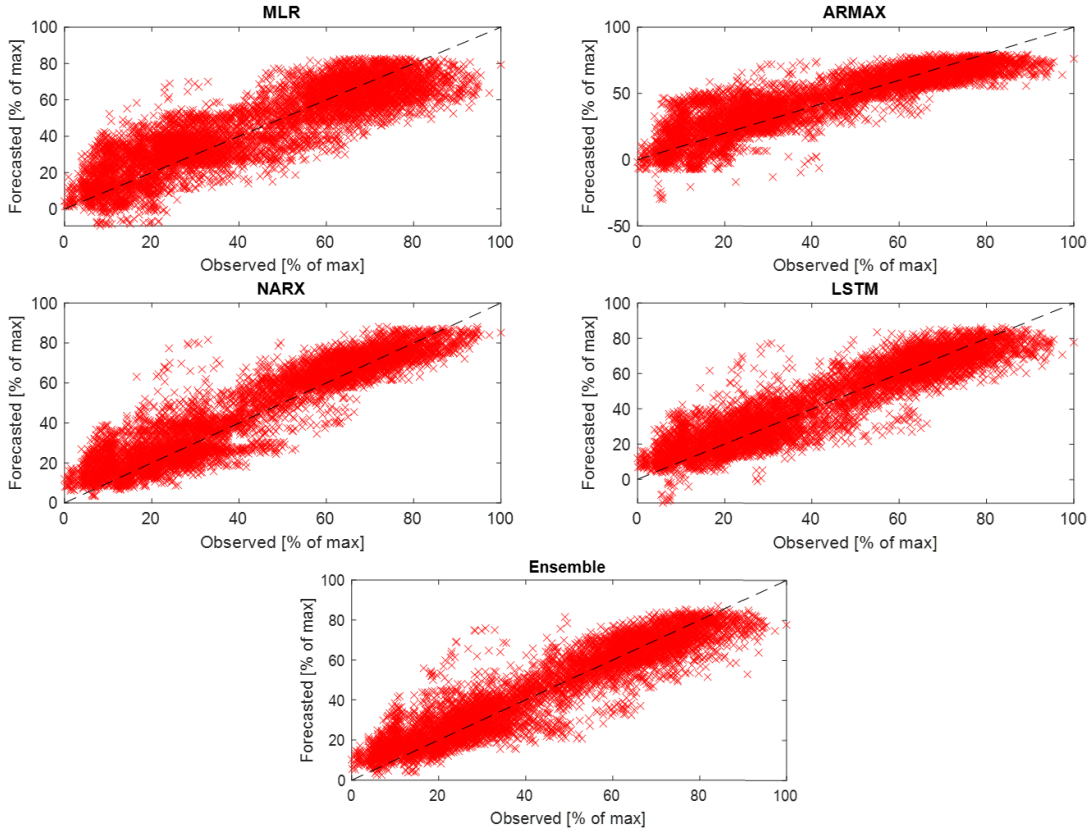

Figure 29. Scatter plots of observed versus forecasted values.

Using only artificial calendar-based predictor variables worked sufficiently well in this study. During variable selection phase covering the seasonality was considered in several dimensions. The energy consumption data has clearly seasonal patterns of 24 and 168 hours, but also the 8-hour seasonality could be a variable to be considered. Usually, the work is done in two or three 8-hour shifts per day. This might cause periods with lower production level intra-day and therefore the energy consumption could be lower on those periods. The visible double-seasonality (or triple-seasonality) can be handled with multiple ways but the selected approach with Fourier-terms turned out to

be effective. For the ensemble model, the SARMA component was explaining the weekly seasonality. A double seasonal ARMA model could be another possible way to catch the double seasonality.

Encoding the cyclical variables such as day of the week or month turned out to be applicable practice. Also, as weekends and national holidays looked to be plainly different compared to production days, it was necessary to do categorical predictors to separate these, and similarly for the hour of the day. Although, with more precise knowledge of the production processes, production schedule, and other relevant variables, the consumption profile could have probably been explained much more specifically.

If a forecast model needs to be implemented with little or no history data available, after a while of data collection a simple method such as hourly average for each day of the week or a simple statistical model such as ARMA, could be considered. Obviously, this would not lead to accurate forecasts, but it may give a picture of the future energy consumption early on.

Overall, the ensemble model showed signs to outperform the other model structures on week-ahead forecasting scenarios. Therefore, it was selected to be tested with the energy consumption data of factory B.

7.1.2 Case 2: Factory B

Similar simulation scenario was run for the energy consumption data of factory B. Used model structure was the averaging ensemble model which outperformed other model structures with the data of factory A. Ensemble model was not utilized in a pre-trained manner, instead the model was trained purely and independently with data of factory B due to differences in data overall. As described on Chapter 5.2.2 the data of factory B had inexplicable anomalies which affected the forecasting. First, forecasting was done on raw data resulting in MAPE of 11.11% with the ensemble model. It was clear that the model could not explain these anomalies with the used predictor variables, and it could also be seen from the trend graph (Figure 30).

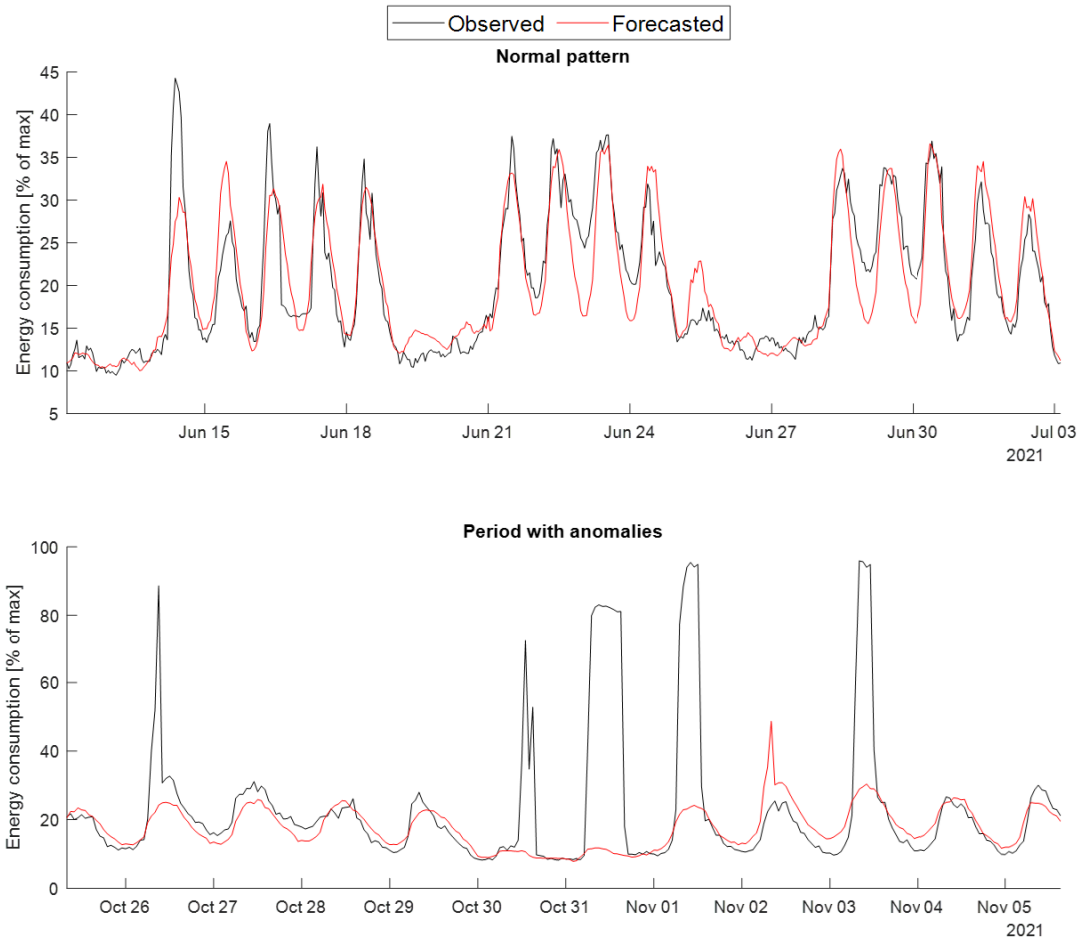

Figure 30. Observed and forecasted patterns (ensemble model) with raw data of Factory B.

As described in Chapter 3.2, the test data used in model validation should also be validated so that the judgement whether the model is valid or not is not based only on the used validation metric, MAPE. Model can also provide poor forecasts if it is used in circumstances which are not considered in model training phase. This is caused by dataset shift which was described in Chapter 3.2. In such situation, the dataset shift type could be for example simple covariate shift or prior probability shift. Therefore, data distribution similarity metrics, for example Kullback-Leibler divergence (Equation 2), Jaccard index (Equation 4), and histogram intersection (Equation 5), were utilized. Figure 31 shows the metrics together with MAPE for each validation. The similarity of the test data was evaluated with these metrics each week against previous 51 weeks of data. For example, on week 43 a vast leap upwards can be seen on histogram distance which indicates that the averages of the tested distributions differ more than on previous weeks. These weeks from week 43 onwards consist of the anomalous data points that were shown on Figure 31. Week 44 of the simulation is an example that if the model validity assessment was only based on MAPE (23.8%), a wrong conclusion would get made because the underlying reason is that the test data was not representative at all.

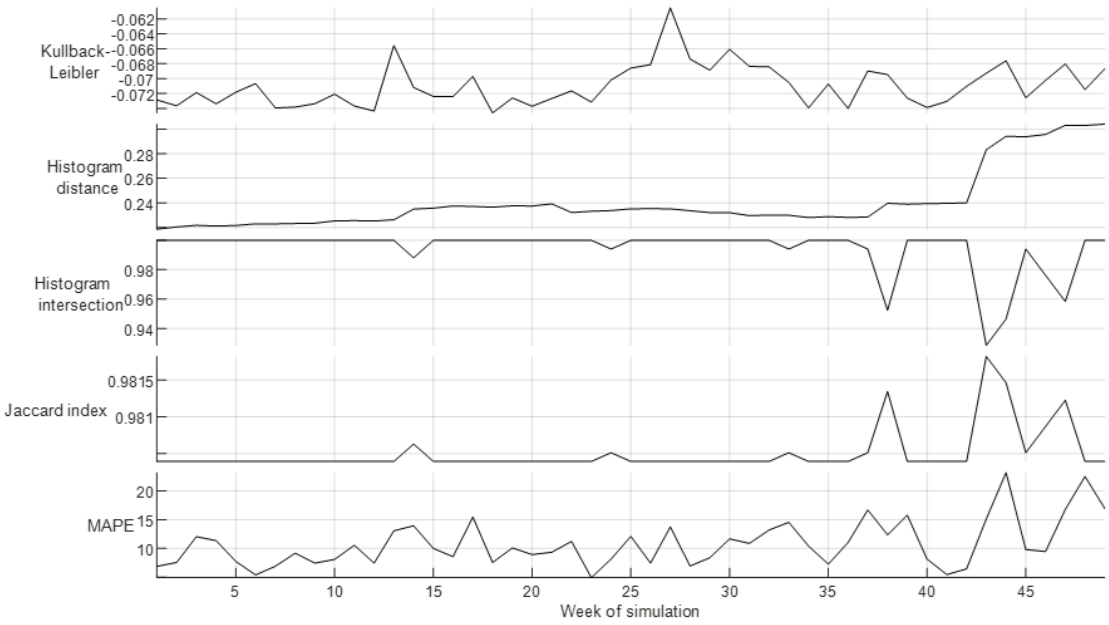

Figure 31. Test data validation metrics (data of Factory B).

To get more proper results of the week-ahead forecasting scenario the last two months of the data were removed. The ensemble model performed better after extracting the poor data, resulting in MAPE of 9.93%. The model was retrained in 50.5% of the weeks. A graph of the forecast profile after data removal is on Figure 32. Notice the difference in y-axis scaling between the normal pattern in Figure 30 and the pattern in Figure 32. The point anomalies on the last two months of data were truly abnormal compared to rest of the data. Still after data removal there were still few point anomalies in the data which left unexplained. This causes that the true maximum energy consumption is probably still lower than the maximum on the processed data. Without no further knowledge of the true nature of these possible anomalies, spikes in data were not removed or handled in any way.

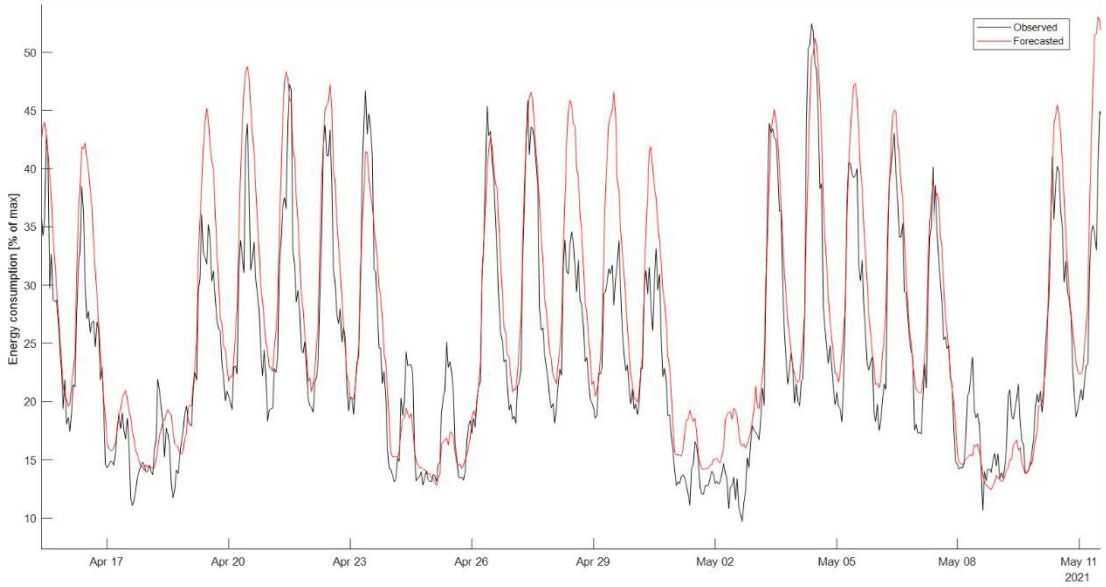

Figure 32. Ensemble model forecast pattern on a three-week period (data of Factory B).

The ensemble model can clearly catch the seasonal patterns of 24 and 168 hours. Nevertheless, the weekend periods seem to cause difficulties for the forecast as the basic level of the profile is changing almost every weekend. Also, from April 28th to 30th there is a visible decrease in the basic level of the daytime energy consumption. Because this was also unexplained in the predictor variables, the forecast could not consider it.

Histogram, normal probability plot, and autocorrelation plots for the forecast error are on Figure 33. Normal probability plot indicates that the tails of the error distribution are extremely long. Some of the forecast errors are very big which suggests that the model has underestimated the energy consumption for that point. That can probably be explained by the point anomalies that were left on the data after the removal of last two months. The ensemble model seems to handle the seasonal patterns of 24 and 168 hours well as the seasonality quickly vanishes in the error autocorrelation plot.

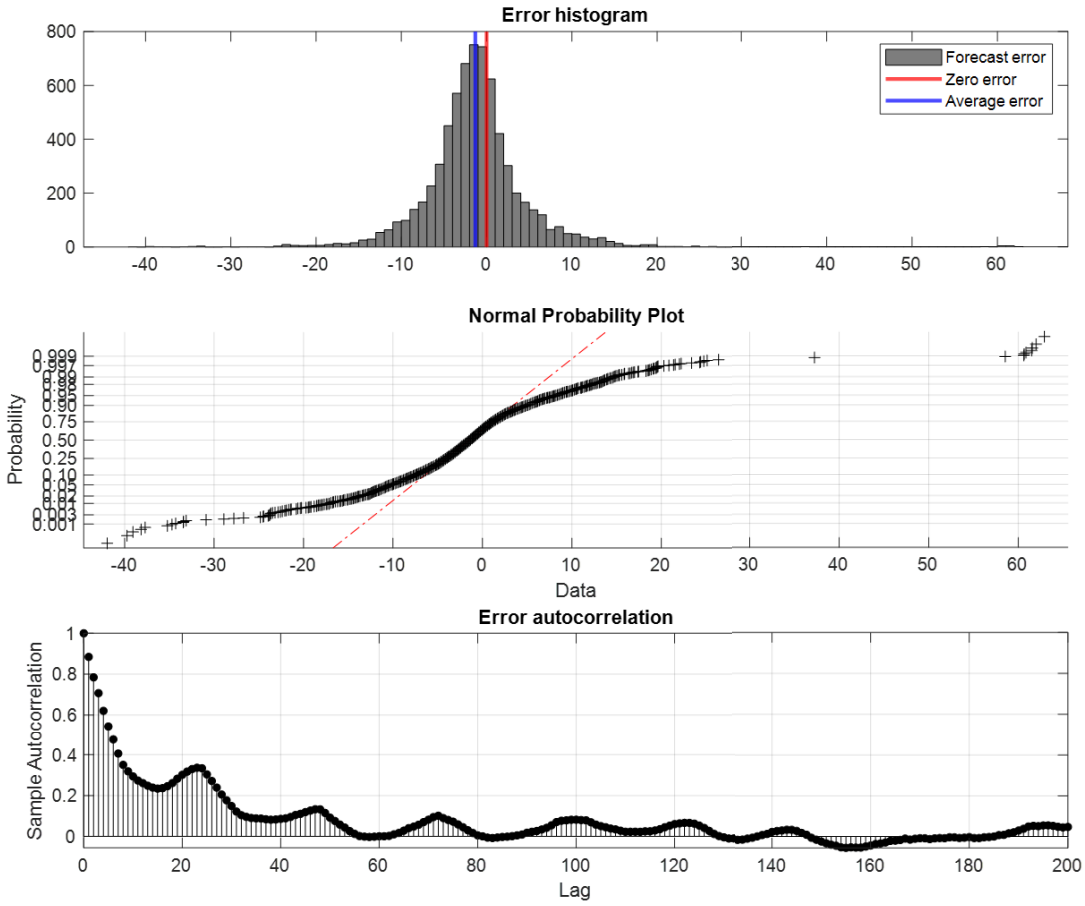

Figure 33. Error analysis for forecast results of factory B.

Scatter plot of the forecast versus actual corresponding observation in on Figure 35. Black dashed line indicates a perfect fit, namely forecasted value would match the observed value exactly. The forecasts match the observations quite closely as most of the points lay around the line representing a perfect fit. As the earlier discussion about the potential anomalous periods suggested, some of the observation with big value were greatly underestimated in the forecast. Also, some of the observations with small value were greatly overestimated in the forecast. The overestimated data points are probably mostly weekends and various weekdays where the energy consumption was less than the forecast suggested. For example, Figure 34 shows differences between weekdays and weekends, and the source of the difference was unexplained. There was no clear pattern visible to help understand which weekdays or weekends have smaller energy consumption than others. Probably the cause is explained by changing production operations and variations in production schedule, and thus the designed artificial calendar-based predictor variables could not explain those days.

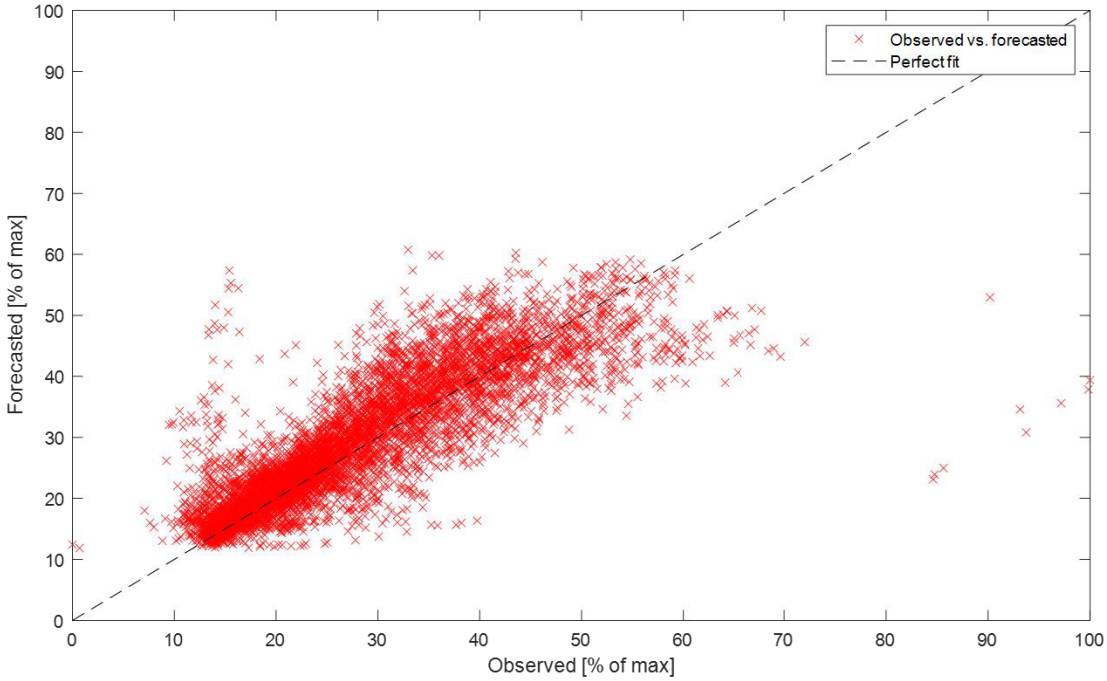

Figure 34. Scatter plot of observations versus forecasted value.

Few other probably relevant hourly average variables were available for factory B: total district heating consumption, total water consumption, and outside temperature. Cross-correlation between the forecast error and each of the other available variables was calculated with lags of ±200 hours. The cross-correlation plots are shown on Figure 35. Based on the cross-correlation plots, every variable has slight correlation with the forecast error. Forecast-wise utilizing the outside temperature looked promising. Outside temperature seems to explain the forecast error at some level, especially with negative lags. This means that outside temperature at time t could help explaining the energy consumption in the future, to some degree. Outside temperature could also be useful as a predictor variable in time-series forecasting as the value of the predictor variables must be known or predicted for the forecast horizon. In comparison, the amount of consumed district heating or water might be hard to be predicted. Then again, outside temperature forecasts can be found from public sources, even for no costs for some locations. The weather forecasts queried from public databases must be merged into correct format in order to use them in week-ahead forecasting. History data for weather forecasts could not be found from public sources for the needs of this study.

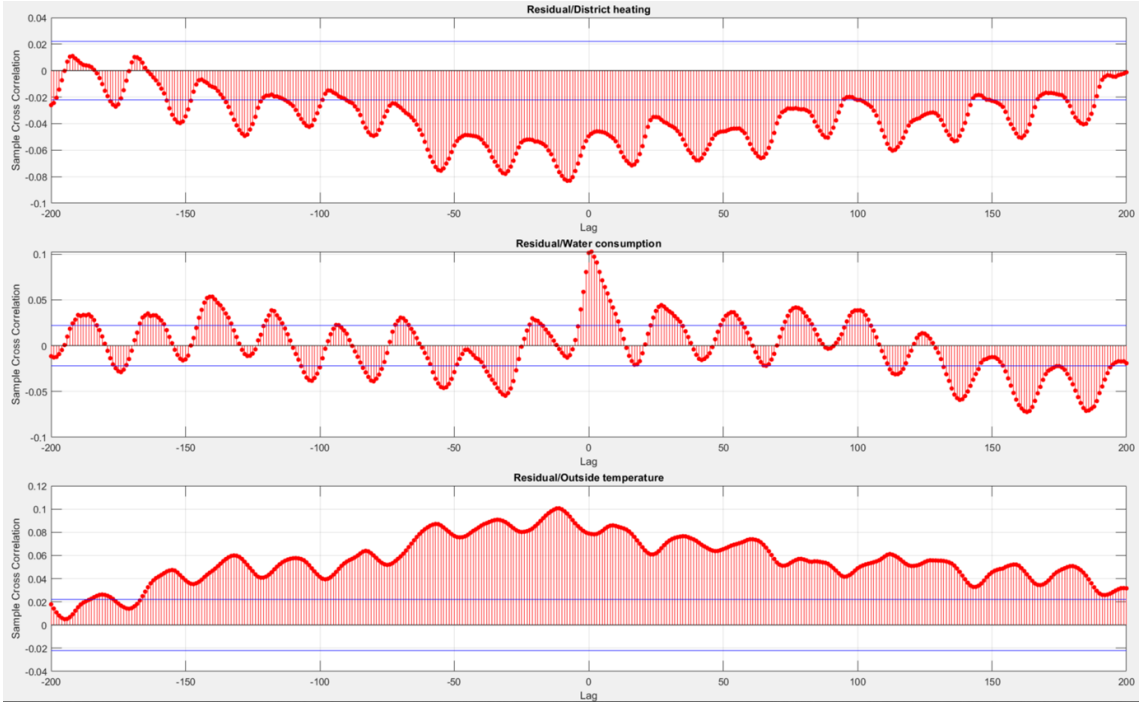

Figure 35. Cross-correlation plots between model residual and studied variables.

Therefore, outside temperature was tested as a predictor variable by using actual measured hourly average data queried from time-series database. This approach is problematic because the temperature used for the forecast horizon is actual precise value without the uncertainty of the weather forecast. The interest itself in this experiment was to see whether the outside temperature affects the forecast accuracy. Inserting the outside temperature to predictor variables was tested on the original raw data of factory B. As told earlier the original forecast scenario with ensemble model resulted in MAPE of 11.11% whereas the ensemble model enhanced with the outside temperature resulted in MAPE of 10.93%. Hence, the forecast accuracy improved subtly. On the other hand, outside temperature might not be the most relevant variable to consider for factory B. Nevertheless, adding the outside temperature shall be considered case by case during the variable selection phase.

7.1.3 Discussion on the week-ahead forecasting

The baseline energy consumption was identified with time-series forecasting methodologies. The idea behind using time-series forecasting for this purpose was to identify a baseline profile for the forecast horizon period, the next week in this case. Energy consumption forecasting is important in order to be able to handle the energy balance of the organization more efficiently. By knowing a baseline for the future week, it is easier to optimize the energy usage and related costs. Also, the baseline helps to consider whether the current energy consumption is lower or higher than it should be.

The forecast error could be utilized as a metric to define the impact of actions regarding continuous improvement in processes. Hamedi & Mokhtar (2019) proposed the cumulative sum of the model residual which could be used for example when calculating the savings of a change or improvement in operating practices. What is important to consider is the difference between the standard deviation of the model residual and the value of the savings. It is necessary to distinguish how big part of the residual can come from the natural variance of the model estimate. For example, for the averaging ensemble model the standard deviation of the residual was 8.2% of the maximum energy consumption. This means that for example if our maximum energy

consumption is 1000 kWh, then the standard deviation of the residual would be 82 kWh. Then if we calculate the difference between the measured consumption and the forecasted consumption and get for example an energy consumption savings value of 150 kWh, we have to consider how much of the savings value is actually saved because some of the savings can also be explained by the natural variance of the model. Similarly, if the baseline forecast is used to detect upwards shifts, we have to consider the standard deviation in that case as well. Figure 36 illustrates the idea how the baseline model could be used in order to evaluate metrics from model residual. The grey area around the forecast indicates the confidence interval of the model residual. The boundaries of the interval are calculated by adding and subtracting the standard deviation of the residual to the forecast. As seen from the figure, the area between the baseline and the improved energy consumption pattern is actually smaller when the model variance is considered as well.

The energy consumption baseline identification approach of Hamedi & Mokhtar (2019) was a bit different as it did not implement a dynamical time-series forecasting based baseline identification procedure. Also, the set of input variables in their studies was more versatile, including process variables such as production rate, number of plant's start-ups and shut-downs, and cooling degree days. The MAPE of their best performing

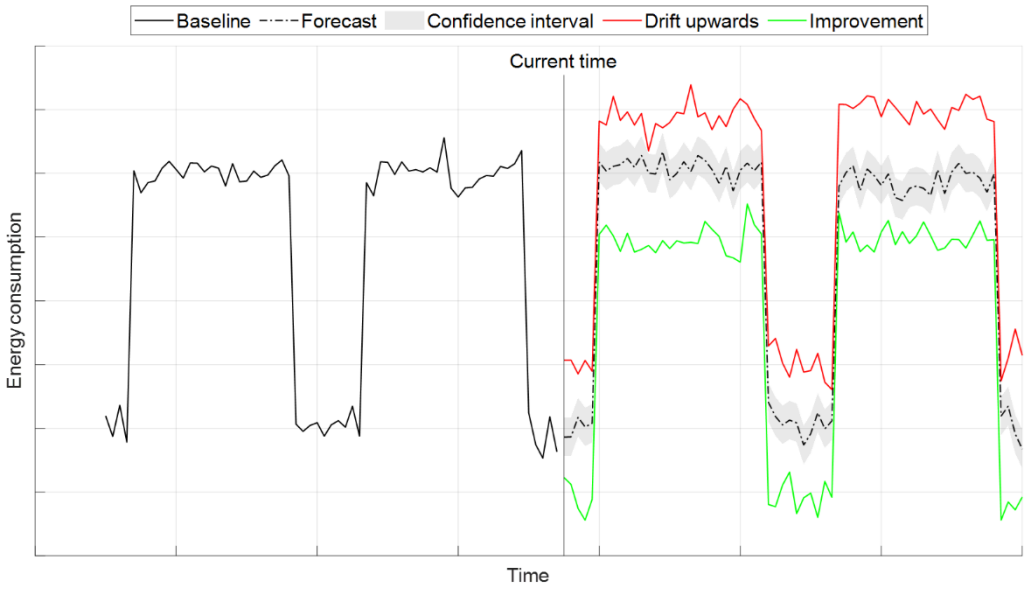

Figure 36. Example: Baseline with comparison to two different scenarios.

model, MLP-ANN, was 2.1% for electricity consumption which is surely much better than the MAPE achieved in this study. On the other hand, it is more than expected that the modelling error gets bigger on a multi-step time-series forecast approach than in a simpler estimation problem.

The averaging ensemble model seemed promising also for the data of factory B. There were some difficulties with unexplained highly anomalous data points in the raw data of factory B. This caused that the forecast was totally off for those periods and therefore it had a significant impact on the forecast accuracy. The averaging ensemble model resulted in MAPE of 11.11% with the raw data. After the anomalous periods were removed from the data, MAPE reduced down to 9.93%. As per forecast accuracy, this is close to the result the ensemble model achieved with the data of factory A. For factory B also few other time-series variables, district heating consumption, water consumption, and outside temperature, were studied subtly. The interest was whether these variables correlate or cross-correlate with the model residual. The variables seemed promising cross-correlation-wise since with certain lags there were correlations between the model residual and other variables. For further studies the outside temperature could be considered in a more detailed level. The outside temperature as a predictor variable improved the forecast accuracy. The problem with the studied case was that the outside temperature was an exact measurement instead of a forecast into the future. This was because weather forecasts for outside temperatures were not available for the examined periods in the past.

As a future note, the relevant variables should be considered with more domain expertise available in order to identify a more robust and trustworthy model even though the forecast accuracy appeared to be good. Also, data mining approaches could be interesting as usually industrial time-series databases can have at least hundreds or even thousands of variables available. Highly optimized processes are probably already taking most of the relevant variables into account but with data mining even more underlying relationships could be found. In this study the variable selection methods (such as forward selection presented in Chapter 5.3) were not utilized because there were not enough variables for that purpose. A possible approach could be that all the

available and applicable variables were evaluated and after some screening the size of the set of variables would be convenient for more detailed studies.

Model complexity and interpretability are important concepts when considering a model to be used by users who are not so familiar with machine learning approaches. For example, the LSTM model which proved to be an effective model structure for time-series forecasting is a really complex deep learning model, and thus its structure is very hard to be interpreted. As an alternative, an MLR or an ARMAX model can be much easier to understand and still in some cases they can be capable to produce week-ahead forecasts with sufficient forecast accuracy.

One possible way to lessen the need of training data and training resources in model implementation phase would be to use pre-trained models. However, finding pre-trained machine learning models for energy consumption time-series forecasting purposes might be difficult if not impossible. In spite of that, if uniform data from similar kind of facilities was available, pre-training an energy consumption forecasting model with massive amount of training data could be a valuable approach. As stated, in this study only the model structure for factory B was decided based on the experiments with the data of factory A.

The simulation scenario in these experiments started with an initial dataset of 52 weeks. In real life application the model implementation is probably wanted to be started earlier than after 52 weeks of data collection. For this purpose, it is suggested to start with more simpler model structures such as hourly average per weekday or with a statistical model such as ARMA, which may not need as much data as the better performing deep learning methods. The forecasts made by models trained by small training datasets might not perform very well but they may still give valuable information for the organization early on.

The importance of carefully executed exploratory data analysis cannot be underestimated. One possible way to simplify the needed model structure would be to identify different operating conditions as precisely as possible and deploy multiple

models for different conditions. In these separate conditions fairly simple models such as MLR could probably also perform just as well as a complex deep learning models do. With the studied data, separate models could be identified for example for operating hours on weekdays, non-operating hours on weekdays, daytime on weekends, and nighttime on weekends.

The standards related to the topic, ISO 50001 and 50006, propose quite vague guidance on model-based energy baseline identification approaches. For example, the ISO 50006 which is a supplementary extension to the ISO 50001, does not really suggest how the model should be identified and which model structures should be considered. Despite the fact that there are standards behind the concept of energy baseline, the vagueness of the concept leaves much obscurity for the baseline identification and implementation process. On the other hand, as the scope of different possibilities in data-driven modelling is really wide, it is naturally hard to capture everything under standards. The ISO 50001 and 50006 standards seem to provide quite little guidance or regulations on the modelling, therefore leaving room for data engineering solutions.

7.2 Time-series anomaly detection

Time-series anomaly detection for energy consumption data was studied with three approaches: statistical process control, isolation forest algorithm, and autoencoder.

7.2.1 Statistical process control

Statistical process control (SPC) experiment was carried out in a rolling window of 12 weeks of daily average energy consumption data. The mean and standard deviation to be used when deciding the limits and corresponding control zones for the upcoming week were calculated from the data of previous 12 weeks. The limits and control zones were kept steady for the whole upcoming week (5 days, weekends excluded) and therefore the limits appear in a staircase manner. Figure 37 shows the SPC experiment with applying the rules described in Chapter 6.6.1. Green circles indicate the randomly added point anomalies, and vertical lines “Shift 1/2/3” indicate the added positive bias

shifts. The figure and the rule violations in it are read so that the indicator of a violation is on the data point which triggered the rule. For example, Nelson 3 rule which is applied in order to identify unusual trends, triggers when the sixth consecutively increasing or decreasing data point is recorded. Hence, every following data point can also trigger the violation if the rule violation remains.

Nelson 2 rule appeared to cause most violations for the used data with 73 hits. Also, Nelson 1, Nelson 3 and Western-Electric 2 rules got 8, 1, and 4 hits, respectively. All the randomly added point anomalies were caught by the used SPC rules. In addition, the added positive bias shifts were captured with Nelson 1 and Western-Electric 2 rules. The raw data itself also triggered violations. For example, on both Julys, 2020 and 2021, there were some data points triggering Nelson 2 rule which is triggered when nine (9) or more consecutive data points lay on one side of the population mean. July is a common month for summer holidays in Finland and therefore lower energy consumption could probably be explained with less employees being in the facility. The production operations run around the year but for example white-collar employees might have been off work for their summer holidays. The energy consumption used in the experiment was the total energy consumption of the whole facility and therefore the consumption of production area and offices could not be investigated separately.

To take a closer look at the SPC rule violations in the raw data, a histogram comparison between the anomalous data and the rest of the data was performed. Comparison is on Figure 38. Histograms differed with clarity. The histogram of anomalous data appeared to be shifted upwards average-wise. The average of anomalous data was 4.5% higher than the average of rest of the data. With this said, statistical process control approach proved its ability to distinguish anomalous data points out of a daily average energy consumption data.

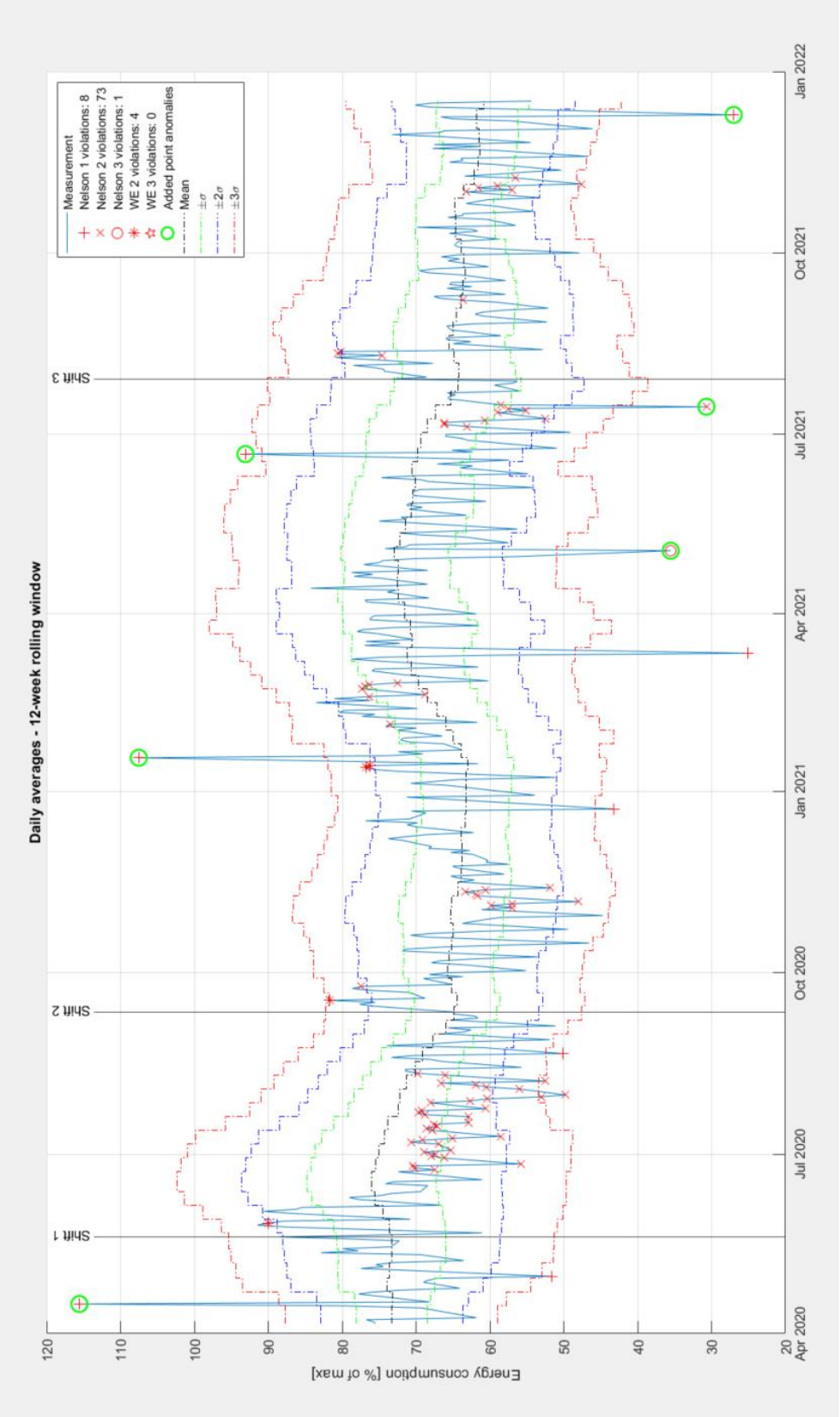

Figure 37. Results of statistical process control approach.

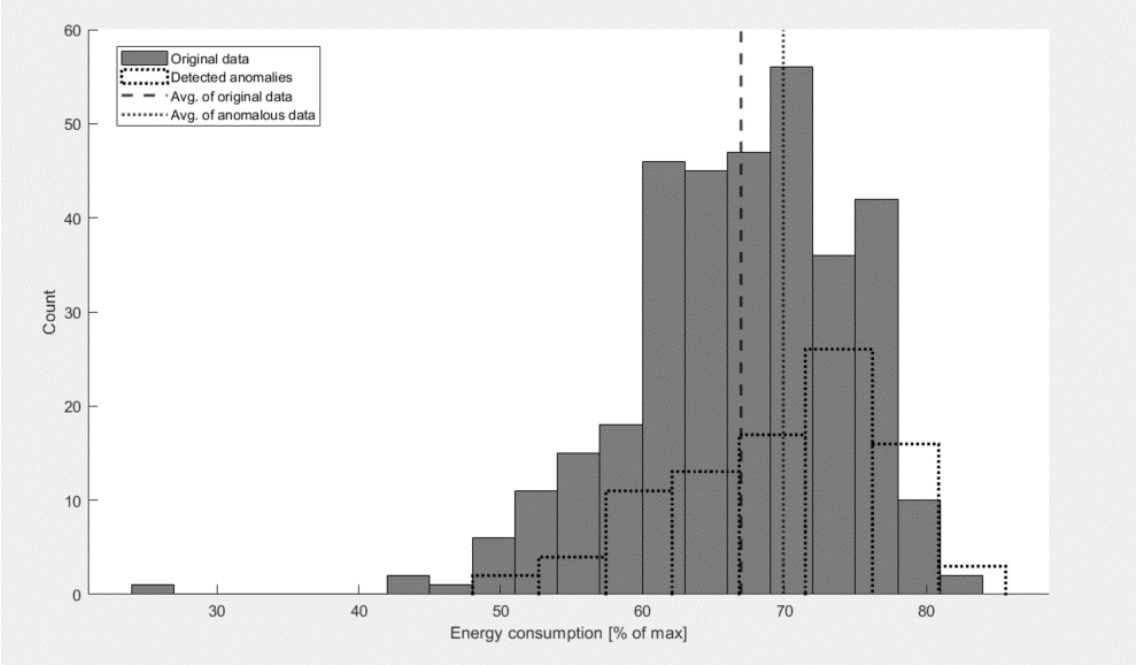

Figure 38. Comparison of original and anomalous data.

7.2.2 Isolation forest

First half of the used data was utilized for training of the isolation forest, and rest as a test data. Figure 39 shows the results of isolation forest anomaly detection experiment. Training and test data division is indicated with a black vertical line. Isolation forest algorithm seems to detect the added point anomalies moderately well. In addition, the algorithm detected anomalies that were on the original raw data. For comparison, blue and red dashed lines indicate the ± 2 and 3 sigma limits of the training set, respectively. Almost all of the detected anomalies appear to be beyond three-sigma limits. Point anomalies were added with a probability of 0.2%.

It is important to select the contamination fraction for the isolation tree properly. Figure 40 shows a situation where 0.4% of the data was anomalous upwards or downwards, but the contamination fraction was only 0.2%. Therefore, the number of anomalies in the training data was twice the number that was actually defined in the training phase of the isolation tree. Thereby, the anomaly detection accuracy was poor and lots of true anomalies were left undetected due to wrong decision made in setting of the threshold.

On the other hand, algorithm can be made much more sensitive to outliers but adjusting the contamination fraction bigger. Figure 41 shows a situation where the contamination fraction was 1% but the probability of added point anomalies was only 0.1%. In this case algorithm has detected more anomalies on the raw data itself. Secondly, most of these detected anomalies which were not added by hand look about as anomalous as the self-added ones. There are also few anomalies detected within the three-sigma limits so the threshold might have been set too strict.

For the scenario shown in Figure 41, the histogram comparison between the original raw data and the detected anomalous data is shown in Figure 42. Anomalous data has 89% lower average than the average of the rest of the data.

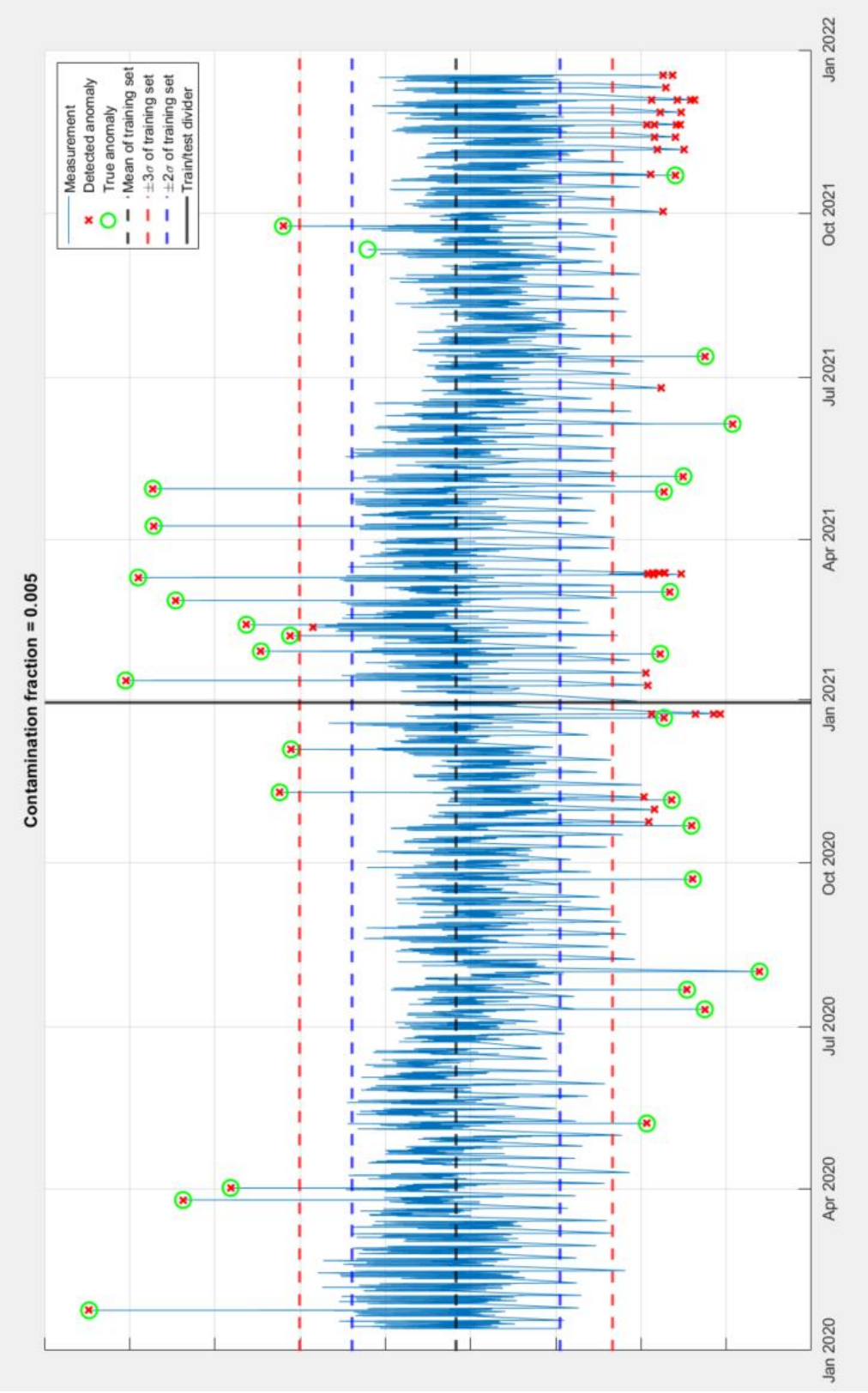

Figure 39. Isolation tree anomaly detection.

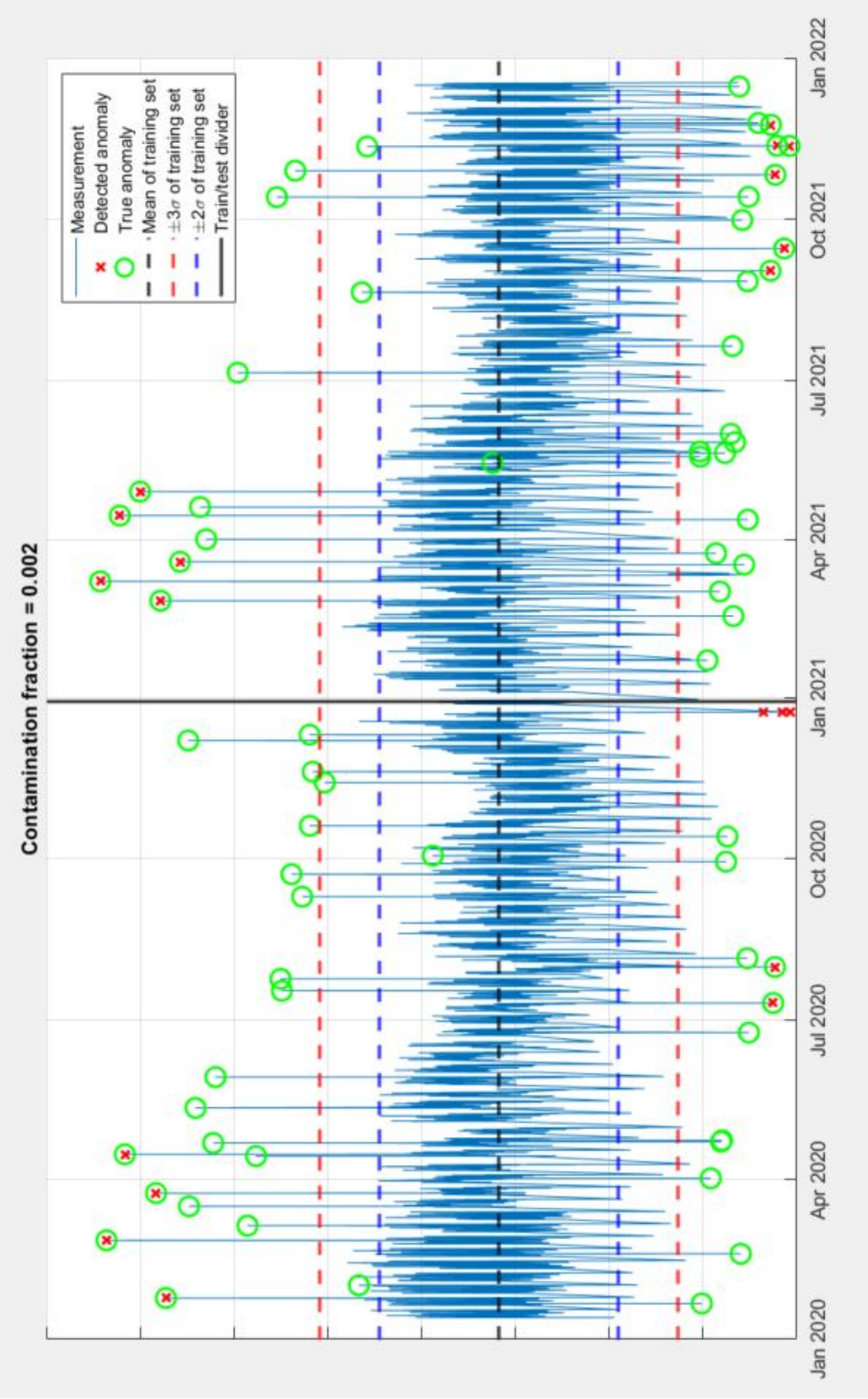

Figure 40. Isolation tree with big contamination fraction.

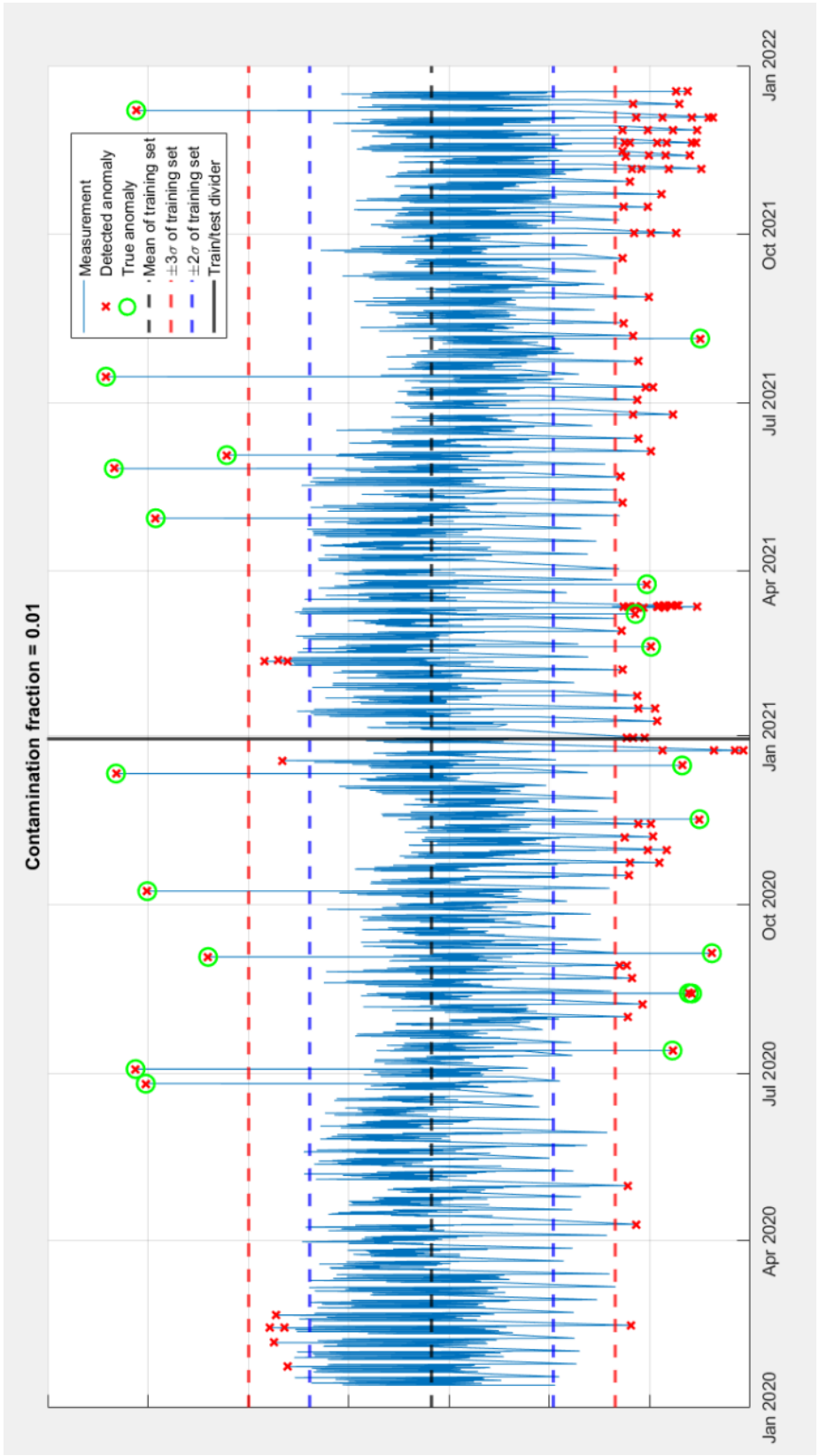

Figure 41. Isolation tree with small contamination fraction.

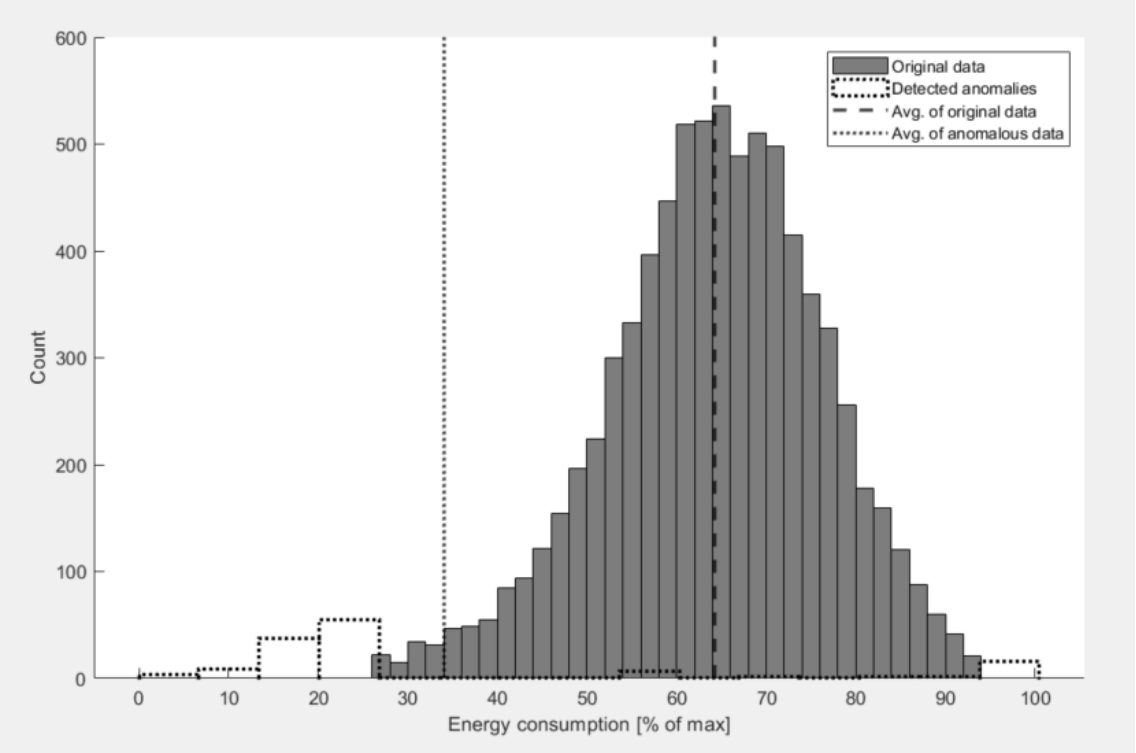

Figure 42. Comparison of original and anomalous data.

The anomaly detection ability of isolation forest algorithm seemed promising with time-series data. Tuning the isolation tree regarding contamination fraction is an essential part in order to achieve an efficient and accurate anomaly detector. Contamination fraction must be considered case by case, and precise labeling of training data is important so that the contamination fraction can be set correctly. In this study without any labeling of anomalies, the scenarios had to be invented with help of a random number generator. This means that also the contamination fraction was set without any domain expertise.

7.2.3 Autoencoder

Because an autoencoder needs lots of training data, it was trained with the energy consumption data of year 2020 and tested with the data of year 2021. The calculated anomaly scores and a histogram of them is in Figure 43. Most of the data has really small anomaly score indicating that those data points do not differ significantly from the training data. Even though the distribution of anomaly scores does not follow normal distribution, three-sigma rule can be applied to make a threshold which points should be considered as anomalies. Setting the threshold is explained in Chapter 6.6.3 (Equation 22 and Equation 23).

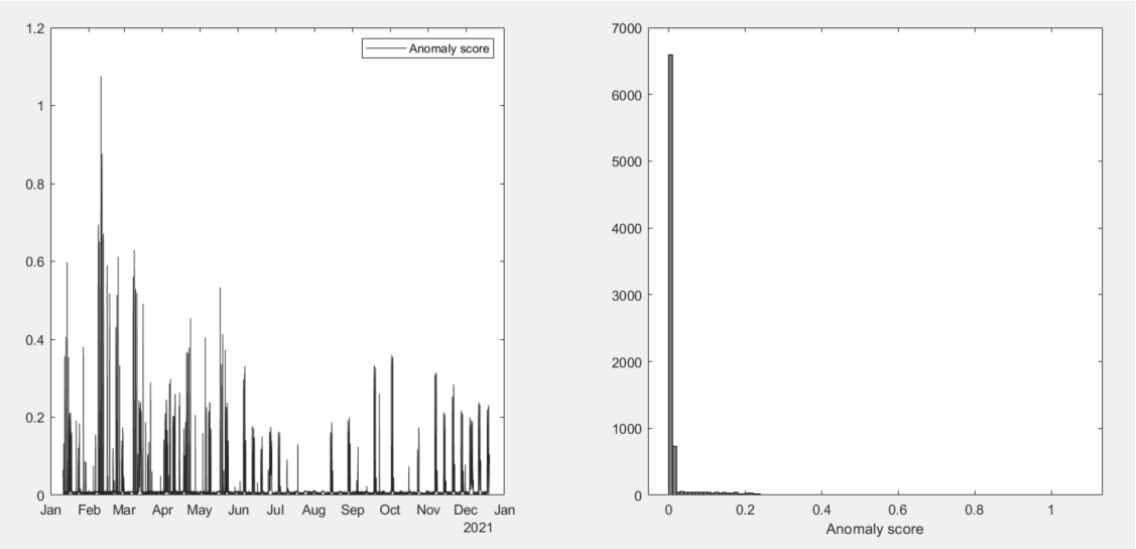

Figure 43. Anomaly scores.

Figure 44 shows the energy consumption of 2021 with the detected anomalies marked with red markers. Autoencoder approach is clearly detecting anomalies which are far away from population average. Using an autoencoder in point anomaly detection could be beneficial but on the other hand training an autoencoder requires a massive amount of data.

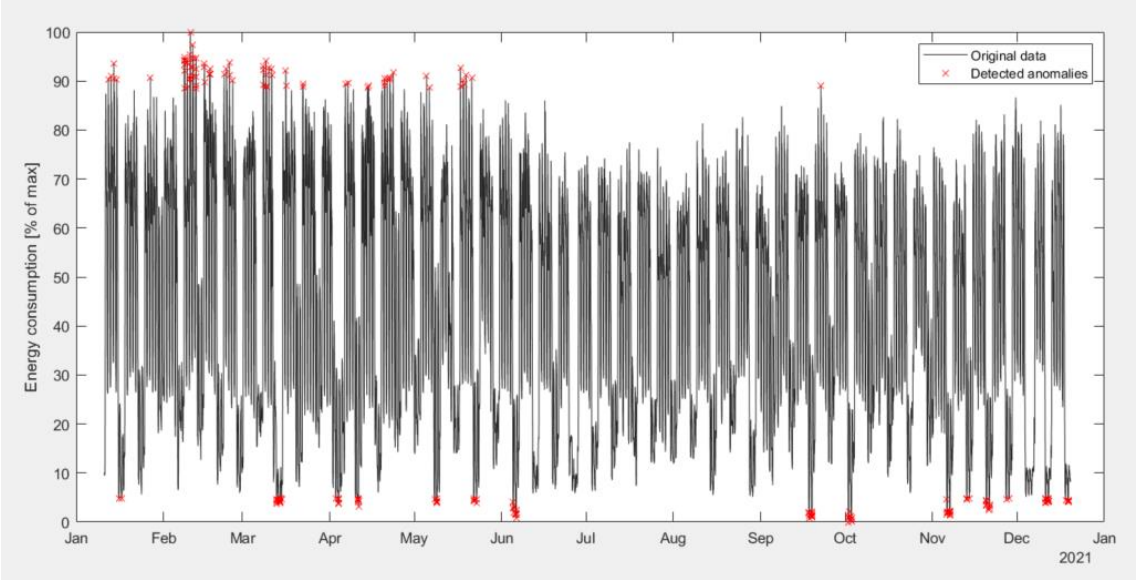

Figure 44. Detected anomalies with autoencoder.

7.2.4 Discussion on the time-series anomaly detection

The experiments with the studied anomaly detection methods were not executed in terms of pure comparison between the methods. Instead, the scenarios were built in order to test them the way they could be expected to be used. For example, statistical process control experiment was designed with an idea to capture longer term abnormal periods rather than anomalies in hourly average data. Therefore, daily averages were used for statistical process control experiments. For isolation forest and autoencoder the hourly averages of energy consumption were used in order to evaluate the shorter-term point anomaly detection capability of those methods. It is important to identify anomalies from energy consumption in order to start a root cause analysis of the possible underlying fault. Possible causes for the anomalies in energy consumption can be for example faults or changes in processes, faults in sensors, or leakages in pipes.

Used data was totally unlabeled in term of anomalies which caused difficulties to interpret the results. For example, common anomaly detection performance indicators such as metrics derived from confusion matrix could not be used. Instead, the distribution of the anomalies detected in the raw data were compared with the distribution of rest of the data. Also, the detected anomalies were interpreted by human

in a contextual way. To add labeled anomalies, some randomly generated point anomalies and positive bias shifts were manually added.

All the methods seemed to work fairly well in their tested purpose. Statistical process control approach is probably the most intuitive and most easily understandable of these three, whereas isolation forest and autoencoder approaches are more complex to interpret on model side. Simplicity of the used method is relevant so that the end-user knows as good as possible what the method is doing and can also detect if the model needs to be enhanced. Overall, a good practice in all modelling occasions is to go with as simple model as possible. The model and its outcome should be interpretable by the users. Not that only the developer of the model understands it and for the rest it is a completely black box.

8 CONCLUSIONS

The profiles of the forecasts with different methods proved that the studied models were able to learn the underlying energy consumption pattern fairly well, with highest MAPE for the validation data of Factory A being 13.8%, and lowest 9.3%. Used predictor variables were purely artificially designed with intention to explain the calendar effect on a strongly seasonal data. In predictor variable design phase there was only limited domain expertise available, so the variables had to be designed only based on exploratory data analysis. Despite that, the predictor variables seemed to perform surprisingly well. An interesting approach to be studied in the future could be to use a wider and more precise knowledge of production schedule and rates. This would widen the set of input variables and make it more descriptive even though the artificial calendar-based variables seemed to provide promising results for a strongly seasonal data like this. Another interesting approach could be to use a simple linear structure with the extended process knowledge in terms of wider set of input variables and more precisely defined operating conditions. Thus, multiple models, one for each operating condition, could be identified. In case of a really wide set of input variables, the most significant ones could be selected using the variable selection methods presented in Chapter 5.3.

Time-series anomaly detection was studied with hourly and daily average energy consumption data. Three approaches were tested in the experiments: statistical process control, isolation forest, and autoencoder. With more detailed tuning of the studied anomaly detection methods even more accurate anomaly detection could be possible. Also, the need for labeling the data cannot be underestimated. To achieve effective anomaly detection in industry, the known changes in the operation of a process should be logged in a systematic way. This could ease the data-driven anomaly detection approaches as the data is in an interpretable form from the beginning. The “hand-written” diary type of notes are always important to describe what was going on. On the other hand, to go through diary type of notes from a long period takes a lot of time and effort in order to use the information in data-driven approaches.

9 SUMMARY

The main objective of the thesis was firstly to find out which machine learning supported modelling techniques are suitable for identifying a baseline energy consumption. Secondly, another big part of the scope was to study time-series anomaly detection methods, and how suitable they would be with energy consumption data. The need of an energy baseline is more and more important for organizations in order to maximize the flexibility of their energy consumption in the fluctuating energy market. It is important to know the state of the energy consumption in the future as accurate as possible. Also, it is important for an organization to know their own energy consumption characteristics in order to reflect the present situation to the past. In this study time-series forecasting methods were studied and evaluated for this large-scale problem.

The thesis includes a literature review of the concepts under the standards ISO 50001 and ISO 50006. Also, the key elements of time-series data and the theory of the exploratory analysis of it was included in the literature review part. Then the theoretical background of data-driven modelling of energy baseline and anomaly detection with time-series data were covered, and the methods were evaluated in the experimental part of the thesis.

The results suggest that the studied energy baseline identification methods can be applicable as they seem to provide good forecast accuracy week ahead. The best performing model structure was an average ensemble model which resulted in a mean absolute percentage error of 9.3% for the validation data of factory A. The ensemble model consisted of three model structures: a SARMA-model, a NARX-model, and an LSTM-model. The model structure was tested also on a data of another factory and the model structure resulted in mean absolute percentage error of 9.93%. The model was re-trained with the data of factory B, so the generalization ability of the model was not tested directly with a model trained on data of factory A. With a large data set a possible approach could be to implement a complex pre-trained model for the needs of multiple similar facilities.

REFERENCES

Arlot, S. & Celisse, A., 2010. A survey of cross-validation procedures for model selection. Statistics surveys 4, pp. 40–79. https://doi.org/10.1214/09-ss054

Baev, P. K., Nicholas M. & Harry T., 2022. Energy Crisis Amidst the Ukraine War: Three Scenarios. PRIO Policy Brief, 7. Oslo: PRIO. Retrieved 11th October 2022, from: https://www.prio.org/publications/13179

Bakirov, R., Fay, D. & Gabrys, B., 2021. Automated adaptation strategies for stream learning. Machine Learning. 110, pp. 1429–1462. https://doi.org/10.1007/s10994-021-05992-x

Box, G. E. P., 1976. Science and statistics. Journal of the American Statistical Association, 71(356), pp. 791–799.

Box G. E. P., Jenkins G. M., Reinsel G. C. & Ljung G. M., 2015. Time series analysis — forecasting and control (5th edition). John Wiley & Sons. ISBN: 978-1118675021

Bruton, K., O'Donovan, P., McGregor, A., & O'Sullivan, D. D. T. J., 2018. Design and development of a software tool to assist ISO 50001 implementation in the manufacturing sector. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 232(10), pp. 1741–1752. https://doi.org/10.1177/0954405416683427

Cui, J., Jin, Y., Yu, R., Okoye, M. O., Li, Y., Yang, J., & Wang, S., 2022. A robust approach for the decomposition of high-energy-consuming industrial loads with deep learning. Journal of Cleaner Production, 349 https://doi.org/10.1016/j.jclepro.2022.131208

Elamin, N. & Fukushige, M., 2018. Modeling and forecasting hourly electricity demand by SARIMAX with interactions. Energy, 165, pp. 257–268. https://doi.org/10.1016/j.energy.2018.09.157

Fingrid Oyj, 2022. Fingrid's electricity system vision 2022 – draft scenarios for the future electricity system [Web document]. Retrieved 11th October 2022, from: https://www.fingrid.fi/en/news/news/2022/future-scenarios-for-finlands-electricity-system-published/

Gers, F. A., Schmidhuber, J., & Cummins, F., 2000. Learning to forget: Continual prediction with LSTM. Neural Computation, 12(10), pp. 2451–2471. https://doi.org/10.1162/089976600300015015

Guyon, I. & Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of Machine Learning Research, 3(7-8), pp. 1157–1182. https://doi.org/10.1162/153244303322753616

Granderson, Jessica & Price, Phillip N., 2014. Development and application of a statistical methodology to evaluate the predictive accuracy of building energy baseline models. Energy, Elsevier, vol. 66(C), pp. 981–990. https://doi.org/10.1016/j.energy.2014.01.074

Hamedi, B. & Mokhtar, A., 2019. Applying multivariate linear regression and multi-layer perceptron artificial neural network to design an energy consumption baseline in a low density polyethylene plant. International Journal of Energy Sector Management, Vol. 13 No. 4, pp. 1133–1148. https://doi.org/10.1108/IJESM-01-2018-0012

Harrou, F., 2020. Statistical process monitoring using advanced data-driven and deep learning approaches: Theory and practical applications. Elsevier. https://doi.org/10.1016/C2018-0-05141-5

Hastie, T., Tibshirani, R., & Friedman, J. H., 2009. The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York: Springer. ISBN: 978-0387952840

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. 2nd Edition, Prentice- Hall, Englewood Cliffs, NJ. ISBN: 0-02-352761-7

Hinton, G. E., & Salakhutdinov, R. R., 2006. Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), pp. 504–507. https://doi.org/10.1126/science.1127647

Hochreiter, S., & Schmidhuber, J., 1997. Long short-term memory. Neural Computation, 9(8), pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hyndman, R.J., & Athanasopoulos, G., 2018 Forecasting: principles and practice. 2nd ed., OTexts: Melbourne, Australia. ISBN: 978-0987507112

Hyndman, R. J., & Koehler, A. B., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting, 22(4), pp. 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P., 2019. Deep learning for time series classification: A review. Data Mining and Knowledge Discovery, 33(4), pp. 917–963. https://doi.org/10.1007/s10618-019-00619-1

Jaccard, P., 1912. The distribution of the flora in the alpine zone. New Phytologist, 11, pp. 37–50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

James, G., Witten, D., Hastie, T., & Tibshirani, R., 2013. An introduction to statistical learning. 1st ed. Springer. ISBN: 978-1461471370

Kingma, D. P. & Ba, J., 2014. Adam: A Method for Stochastic Optimization. 3rd International Conference for Learning Representations, San Diego, 2015. https://doi.org/10.48550/arXiv.1412.6980

Komorowski, M., Marshall, D.C., Salciccioli, J.D., Crutain, Y., 2016. Exploratory Data Analysis. In: Secondary Analysis of Electronic Health Records. Springer, Cham. https://doi.org/10.1007/978-3-319-43742-2_15

Kullback, S. & Leibler, R. A., 1951. On Information and Sufficiency. Ann. Math. Statist., 22, pp. 79–86. https://doi.org/10.1214%2Faoms%2F1177729694

LeCun, Y., Bengio, Y., & Hinton, G., 2015. Deep learning. Nature, 521(7553), pp. 436–444. https://doi.org/10.1038/nature14539

Liu, F. T., Ting, K. M., & Zhou, Z., 2008. Isolation forest. Paper presented at the Proceedings - IEEE International Conference on Data Mining, ICDM, pp. 413–422. https://www.doi.org/10.1109/ICDM.2008.17

MacKay, David J.C., 2003. Information Theory, Inference, and Learning Algorithms. 1st ed. Cambridge University Press. ISBN: 978-0521642989.

MathWorks, 2022a. Deep Learning Toolbox. Web page. Retrieved September 1st, 2022, from: https://se.mathworks.com/products/deep-learning.html

MathWorks, 2022b. Statistics and Machine Learning Toolbox. Web page. Retrieved September 1st, 2022, from: https://se.mathworks.com/products/statistics.html

MathWorks, 2022c. Econometrics Toolbox. Web page. Retrieved September 1st, 2022, from: https://se.mathworks.com/products/econometrics.html

Nelson, L. S., 1984. Shewhart Control Chart - Tests for Special Causes. Journal of Quality Technology, 16(4), pp. 237–239. https://www.doi.org/10.1080/00224065.1984.11978921

Nelson, L. S., 1985. Interpreting Shewhart X control charts. Journal of Quality Technology. 17(2), pp. 114–116. https://doi.org/10.1080/00224065.1985.11978945

Nontapa, C., Kesamoon, C., Kaewhawong, N., & Intrapaiboon, P., 2020. A new time series forecasting using decomposition method with SARIMAX model. Neural Information Processing. ICONIP 2020. Communications in Computer and Information Science, vol 1333. Springer, Cham. https://doi.org/10.1007/978-3-030-63823-8_84

Oakland, J. S., 2003. Statistical process control (5th ed.). Butterworth-Heinemann. ISBN: 978-0750657662

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N.D., 2009. Dataset shift in machine learning. MIT Press. https://doi.org/10.7551/mitpress/9780262170055.001.0001

Ren, Y., Zhang, L., & Suganthan, P. N., 2016. Ensemble classification and regression-recent developments, applications and future directions [review article]. IEEE Computational Intelligence Magazine, 11(1), pp. 41–53. https://doi.org/10.1109/MCI.2015.2471235

SFS-ISO 50001, 2018. Energy management systems – Requirements with guidance for use. Finnish Standards Association SFS: pp. 40.

SFS-ISO 50006, 2015. Energy management systems — Measuring energy performance using energy baselines (EnB) and energy performance indicators (EnPI) — General principles and guidance. Finnish Standards Association SFS: pp. 36.

Swain, M. J., & Ballard, D. H., 1991. Color indexing. International Journal of Computer Vision, 7(1), pp. 11–32. https://doi.org/10.1007/BF00130487

Wang, Z., Wang, Y., Gao, C., Wang, F., Lin, T., & Chen, Y., 2022. An adaptive sliding window for anomaly detection of time series in wireless sensor networks. Wireless Networks, 28(1), pp. 393–411. https://doi.org/10.1007/s11276-021-02852-3

Western Electric Company, 1956. Statistical Quality Control Handbook.