Addressing Missing Smart Meter Data in Electricity Consumption Using Machine Learning

Long Short-Term Memory for Enhanced Electricity Consumption Forecasting

Master Thesis submitted to the Faculty of Economics and Business

Andrea JOHN

16718918

Information Management Institute

University of Neuchâtel

For the degree of Master of Science in Applied Economics

Supervised by

Prof Adrian Holzer, University of Neuchâtel

Dr Vladimir Macko, University of Neuchâtel

Neuchâtel, JUNE/2023

Contents

Abstract: This study investigates the potential of Long Short Term Memory neural networks to estimate smart meter electricity daily consumption data for a given household. Long Short Term Memory, LSTM for short, is a machine learning solution particularly well suited for building predictive models in a time-series context, due to its architecture and long-term memory capability. In the first step, the smart-meter electricity data undergoes a comprehensive data analysis, aimed at effectively preparing the data for training. In the second step, a feature engineering approach is applied to add relevant information such as the weekday, temperature or the occupancy of the household to the dataset. A univariate LSTM model, which utilizes past energy consumption data, alongside a multivariate LSTM model, that incorporates the engineered features is designed and trained. Finally, the results of the univariate LSTM model, the multivariate LSTM model, and a baseline method are evaluated and compared.

Keywords: missing values; smart meter data; long short term memory; feature engineering; time series forecasting

1 Introduction

Climate change, a pervasive and persistent global challenge, has led to the establishment of carbon emission regulations through international agreements such as the Kyoto Protocol and the Paris Agreement (Cullen, 2011; Horowitz, 2016). Consequently, these agreements have clearly underscored the increasing importance of strategies focused on the use of renewable energy and the development of efficient power grids. Given that energy consumption within buildings account for over 40% of global energy use - a figure that is steadily rising (Hwang et al., 2020) - improving energy efficiency in buildings has become a crucial aspect of climate change mitigation efforts. At the same time, as the proliferation of innovative smart devices continues, they offer significant potential to influence decision-making processes and promote engagement and interaction (Holzer et al., 2020). In Switzerland, a widespread rollout of smart meter devices, which record the consumption of electric energy, is planned for 2027 (Ecoplan, 2015). It is therefore important to use these opportunities to develop efficient measures to save energy. The work topic presented in this study falls within the framework of the INFINEED project, a joint project involving the three academic departments of Economics, Behavioral Sciences, and Information Systems at the University of Neuchâtel and the University of Applied Sciences and Arts Western Switzerland. INFINEED, which stands for the interplay of feedback and incentive effects on electricity demand, deals with implementing and conducting field and choice experiments to investigate the impact of soft interventions on residential electricity consumption (Holzer et al., 2022). To determine the effect of the various measures on electricity demand, smart meter data from at least 400 households will be used. In order to be prepared for the case of data readout failure in the field experiment, this study conducts analyses of historical records of smart meter data and aims to develop a predictive model for missing values based on the available readings and also to address the problem of missing data in the historical consumption data. To predict the energy consumption one day in the future, a framework of a

long-short term memory model is suggested. In order to compare the performance of the proposed LSTM method, a benchmark method using past averages on the same weekday is applied.

The contributions of this study are:

1. Application of LSTM in the field of electricity consumption forecasting and evaluation of achieved performance.
2. Exploration of the potential impact on forecast accuracy when introducing different features into the dataset.

The content of this study is organized into eight chapters. Following the introduction, the second chapter provides a literature review whereas the third chapter details the methodologies and framework implemented. The fourth chapter introduces the data used for this study and Chapter 5 is focused on outlining the technical configuration, such as the training and validation of the model and the error metrics applied. In Chapter 6, the performance of the models is thoroughly examined and discussed in relation to the benchmark method. Chapter 7 addresses limitations of this study and identifies areas for future exploration. The final chapter provides conclusions from the study.

2 Literature review

In order to choose the most appropriate way to deal with missing values, it is necessary to explore why these missing values arose in the first place. Little and Rubin (2019) identified three distinct mechanisms that can result in missing values. Data can be missing completely at random (MCAR), which is the case if the absence of data is independent of the values of both observed and missing data. In such cases, imputation techniques can be applied to fill in the missing values without introducing any systematic bias or distorting the statistical properties of the dataset. Secondly, data can be missing at random (MAR), where the absence of data is dependent on the observed data. Lastly, data can be missing not at random (MNAR), where the absence of data depends on the values of the missing data itself. Following Wang et al. (2021), missing values in a smart meter data context are of the type MCAR, as the absence depends on force majeure such as facility issues or environmental factors.

The existing literature offers numerous examples of how to handle missing values in time series datasets (Afrifa-Yamoah et al., 2020; John et al., 2019). Statistical approaches but also machine learning approaches are used to fill in missing values (Wang et al., 2021). Broken down to time series studies with an energy context, examples include those in the area of weather data (Kim et al., 2019), wind speed data (Martinez-Luengo et al., 2019), or smart meter data (Peppanen et al., 2016). The latter have developed a data imputation algorithm that outperforms simple methods such as linear interpolation or historical averages. Their method is a combination of imputation by linear interpolation and historical averages, which are weighted differently depending on the scenario. The rationale behind this methodology stems from the understanding that short missing intervals have large similarities with the surrounding values, while on the other hand load data strongly depend on habit patterns. Wang et al. (2021) assessed in their paper *Towards missing electric power data imputation for energy management systems* the performance of different missing value imputation techniques. They concluded that machine learning methods outperform

statistical methods in general, however, the performance is depending on whether the missing value occurs during off-peak (Monday to Saturday from 12:00 pm to 7:30 am and 10:30 pm to 24:00 pm and all of Sunday) or peak times (weekdays from 7:30 am to 10:30 pm). When missing values happened during peak times, machine learning methods are more efficient whereas, during off-peak and semi-peak times, the linear interpolation provided more accurate predictions.

Data imputation techniques are closely connected to the field of short-term load forecasting (Peppanen et al., 2016). A significant amount of recent research has been devoted to exploring diverse methodologies for predicting future data values and conducting comprehensive comparisons among different approaches (Durand et al., 2022). Explored methods include traditional statistical-based models such as Holt-Winters (Zheng et al., 2018) and ARIMA (Guarnaccia et al., 2017) or machine learning-based techniques such as Support Vector Machine (SVM) (Fu et al., 2015) or Neural Networks (Zhang et al., 2018). In a comparative study of different forecasting strategies for energy consumption in smart buildings, Divina et al. (2019) concluded that machine learning-based approaches are particularly more suitable. Lin et al. (2020) developed an LSTM model for electricity consumption prediction, resulting in a notable 6.5% increase in performance compared to the state-of-the-art model. Similar results were obtained by the work of Somu et al. (2020).

Nonetheless, in short-term load forecasting studies, focus lies predominantly on predicting the total system load (Peppanen et al., 2016). This is noticeably different from the focus of this study, the prediction of missing values for each household. Wang et al. (2021) also acknowledges that the imputation of missing values in electricity data has been an area that has received insufficient attention. Therefore, this study aims to contribute to extent the existing literature and provide valuable insights into potential improvements in the accuracy of electricity consumption predictions at the household level necessary for the INFINEED project.

3 Methodology

In the following subsections, the methods used in this study are presented. In Subsection 3.1 Univariate LSTM Model, the underlying technique, Long Short Term Memory Network, is discussed. In Subsection 3.2 Multivariate LSTM Model, it is explained in detail which features are added to the dataset with the aim to improve the performance. Subsection 3.3 Benchmarking baseline delves into the methodology of the baseline model, which is important in order to understand the results of the LSTM models in context.

3.1 Univariate LSTM Model

To first understand long short term memory models, the underlying sequence prediction problem is introduced. Then, classical neural networks are discussed and recurrent neural networks, including the LSTM model, are elaborated.

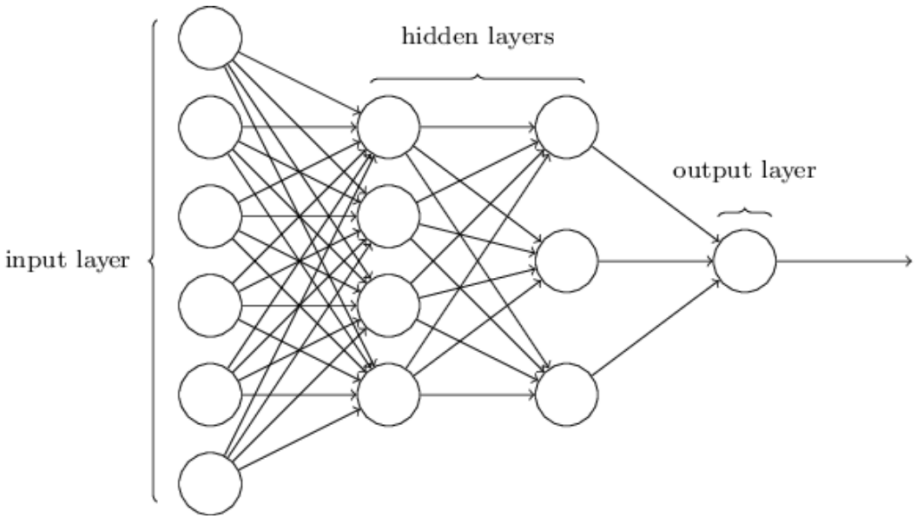
3.1.1 Sequence Prediction

The prediction of the next value for a given input sequence is of interest. Therefore, the objective of sequence prediction is to forecast a value while maintaining the specific order in which the observations of a sequence follow. In distinction to other supervised learning problems where the order of observations may not be significant, sequence prediction necessitates preserving the inherent order of the input sequence (Brownlee, 2017).

3.1.2 Classical neural networks

In the field of sequence prediction, classical neural networks, commonly referred to as multilayer perceptrons or MLPs, play a central role (Brownlee, 2017). These networks are used to approximate the underlying mapping function between input and output variables. Figure 1 shows how such a mapping of input and output is conducted:
The layer on the left side is called the input layer and the neurons (network units) within this layer are called *input neurons*. The layer on the right side is called the

Figure 1 — The architecture of neural networks



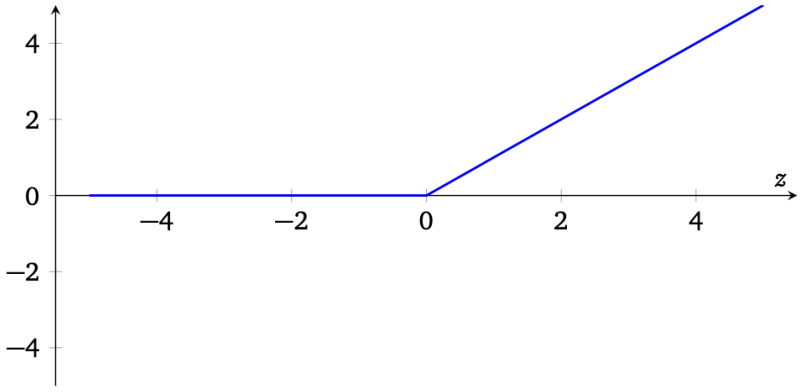
Notes: From *Neural Networks and Deep Learning* (p. 11), by N. Nielsen, 2015, Determination Press. Copyright 2015 by Michael Nielsen. Reprinted with permission.

output layer and contains the *output neurons*, or in this case a single output neuron. The middle layers are called *hidden layers*, in this case there are two hidden layers. The design of the input, hidden and output layers depends on the specific problem to be solved. In the context of this study, when seeking to establish a mathematical relationship between the energy consumption values of the past three weeks and the following day, the input layer consists of 21 neurons and one output neuron. While designing the input and output layers of a neural network is often straightforward, designing the hidden layers can be a challenging task. There are no simple rules of thumb to summarize the design process for the hidden layers (Nielsen, 2015). Therefore, in this study, an existing set-up taken from Brownlee (2018)'s Book *Deep Learning for Time Series Forecasting* is implemented. The chosen set-up is presented in Chapter 5, LSTM Setup.

According to the universality theorem, neural networks consisting already only of one single hidden layer are able to approximate any continuous function to any desired precision (Nielsen, 2015). A hidden layer consists of a set of neurons that perform computations on the input data. Each neuron in the hidden layer applies an activation function to its input that introduces nonlinear transformations to the data. In the case

of this study, a Rectified Linear Unit (see Figure 2), ReLU for short, is used as the activation function.

Figure 2 — Rectified linear unit



Notes: From *Neural Networks and Deep Learning* (p. 123), by N. Nielsen, 2015, Determination Press. Copyright 2015 by Michael Nielsen. Reprinted with permission.

Formally, ReLU is defined as:

$$f(x) = \max(0, w \cdot x + b) \tag{1}$$

where:

- w represents the weight vector ,
- x denotes the input vector,
- b is the bias term.

The ReLU function outputs the maximum of zero and the element-wise sum (scalar product) of the weighted input vector and a bias vector. One of the advantages of ReLu activation functions in a time-series context is the property that negative values are set to zero. In other words, the ReLU function does not activate all neurons at the same time. This makes it possible to focus on the most important features and filter out noise in the data. Also, ReLU mitigates the vanishing gradient problem described in Subsection 3.1.3 Recurrent neural networks.

Neurons in the hidden layer are connected to neurons in the previous layer (input layer or another hidden layer) by weighted connections. These weights determine

the strength and direction of the information flow. Each neuron in the hidden layer usually has an associated bias term that is added to the weighted input before applying the activation function. With reference to the presented neural network in Figure 1, a mathematical representation of one neuron in the hidden layer is given as following:

$$h_i = \text{ReLU}(W_i \cdot h_{i-1} + b_i)$$

where:

- h_i represents the output of the i-th neuron in the hidden layer,
- W_i is the weight matrix connecting the i-th neuron to the previous layer's outputs,
- h_{i-1} is the vector of outputs from the previous layer,
- b_i is the bias vector for the i-th neuron,
- $\text{ReLU}(x)$ is the rectified linear unit activation function.

With regard to the subject of this study, a neural network can establish a mathematical relationship between the historical energy consumption data and the energy consumption on the following day, the day where the value is missing after a sequence of valid values. In other words, neural networks derive estimates about the relationship between input and output variables through a process called *training*. At the beginning of the training process, initial weights and biases have to be determined by the so-called *initialization*. The weights and the biases can be specified e.g. by using independent Gaussian random variables normalized to a mean of 0 and a standard deviation of 1 (Nielsen, 2015). The training data is then passed through the network by each neuron receiving an input, computing a weighted sum of the inputs and including the bias, applying an activation function and passing the output to the next layer of the network. The output of the neural network is compared to the true values of the training set and a value of the so-called *loss function* is calculated. Such a loss function is designed to measure the difference between the predicted output of the network and the ground truth. In order to improve the prediction made, the neural network works with a technique called *backpropagation*. The network

backpropagates the calculated loss through the layers. The goal is to determine how much each neuron, respectively its weight and bias, contributes to the value of the loss function. This process involves calculating the gradients of the loss function with respect to its parameters. These gradients indicate the direction and magnitude of the adjustment needed to minimize the loss. The weights and biases are optimized using an optimization algorithm such as *gradient descent* (Nielsen, 2015). The described process is repeated, usually over several epochs.

Neural networks have a number of advantages over traditional time-series forecasting methods such as ARIMA. ARIMA, which stands for Autoregressive Integrated Moving Average Model, are classical statistical models that can make predictions of time series data given three parameters. The three parameters include autoregression (relationship between an observation and a certain number of lagged observations), differentiation of values to make a series stationary and moving averages, which captures the relationship between the current observation and the residual error obtained from a moving average model applied to previous observations (Brownlee, 2018).

The advantages are briefly discussed below:

- Neural networks do not use strict assumptions about the mapping function and it is possible for them to determine not only linear but also nonlinear relationships.
- Multivariate inputs and also multivariate outputs are possible.

However, the application of classical neural networks in a time series-context is subject to following limitations (Brownlee, 2017):

- Classical neural networks are stateless, which means that it is not possible for them to learn different states. Any output that is dependent on the input sequence is generalized and fixed in the network weights.

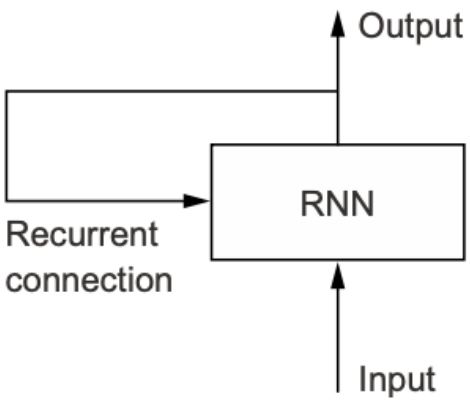
- Classical neural networks do not explicitly understand the temporal structure and it must be illustrated using input modeling.
- Only a fixed size of inputs and outputs is possible.

To overcome these limitations, recurrent neural networks (a.o. Long Short Term Memory) are often used, which are presented in the next subsection.

3.1.3 Recurrent neural networks

Assuming a standard multilayer perceptron (MLP), a recurrent neural network, RNN for short, can be thought of as an extension of this architecture with a built-in loop (Brownlee, 2017) (Chollet, 2021) (see Figure 3).

Figure 3 — A Simplified Recurrent Neural Network



Notes: From *Deep learning with Python* (2nd ed., p. 294), by F. Chollet, 2021, Manning Publications Co.. Copyright 2021 by Manning Publications Co.. Reprinted with permission.

RNNs contain cycles that feed information from previous time steps back into the network as input which can influence predictions at the current time step. This information is stored in the internal states of the network, which enables to store long-term temporal context information (Sak et al., 2014).

RNNs analyse data step by step, processing one observation at a time. In doing so, they learn what previous information is relevant and how it affects future predictions (Brownlee, 2017). However, the main challenge for RNNs is how to train them and adjust the weights properly. It becomes problematic when weight changes cause

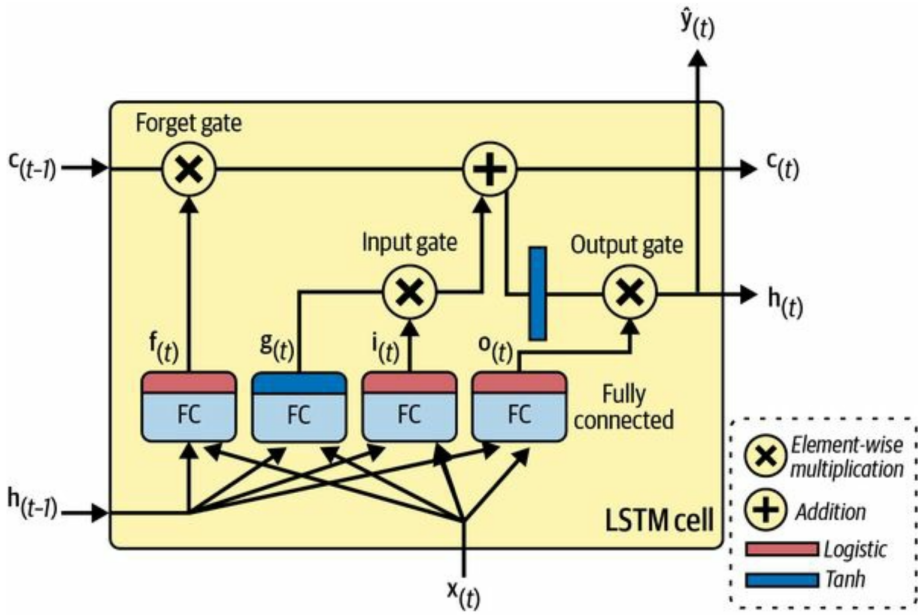
the weights of a RNN to become smaller and smaller until no effect is stored at all (vanishing gradients) or, on the contrary, when they become larger and larger (exploding gradients). Thus, standard RNNs have practical limitations in terms of the amount of contextual information they can access (Brownlee, 2017). In other words, RNNs have difficulties retaining information from previous inputs while processing a sequence. This limitation impedes their capacity to recognize and learn long-term patterns in data, which is in particular of importance in energy consumption data (Géron, 2022). Long Short Term Memory networks are designed to prevent this problem by their architecture which is discussed in more detail below.

3.1.4 Long Short Term Memory

Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture that aims to solve the mentioned short-term memory problem of traditional RNNs (Géron, 2022). LSTM cells excel at capturing long-term patterns due to a special mechanism illustrated in Figure 4. This mechanism consists of three main components: the *input gate*, the *forget gate*, and the *output gate*. This allows the LSTM cells to learn which information to store, which to discard, and which to read from their memory.

The state of the LSTM cell is divided into two vectors: a short-term state $h(t)$ and a long-term state $c(t)$. The long-term state passes through a *forget gate* that selectively erases some information, and then adds new information through an *input gate*. The resulting long-term state is sent without further transformation. This process allows the LSTM cell to store and manage important information over long periods of time. After addition from the *input gate*, the long term state is copied and passed to a activation function. The result of this function is then filtered by the *output gate*, leading to the next short term state. The short term state $h(t)$ is responsible for the new memories. First, the input vector $x(t)$ and the previous short term state $h(t-1)$ are fed into four different layers, which have various functions. The first layer is a gate controller with a activation function. In Figure 4, a logistic activation function is presented. However, in this study, a Rectified Linear Unit (ReLU)

Figure 4 — An LSTM cell



Notes: From *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed., p. 879), by A. Aurélien Géron, 2023, O'Reilly Media, Inc.. Copyright 2023 by Aurélien Géron. Reprinted with permission.

function is employed instead. This layer is responsible for deciding how much of the longterm-state should be erased. The second layer with the output $g(t)$, which is the main layer, analyzes the current input $x(t)$ and the previous short term state. The relevant parts are then stored in the longterm state, while everything else is forgotten. The third layer, the input gate which is controlled by $i(t)$, determines which part of the output $g(t)$ is added to the long-term state. The last layer leads to the *output gate*, which controls which part of the long term state should lead to the next short term state. Such an architecture makes LSTMs particularly effective at handling tasks that run with long-term dependencies. They can learn to recognize important inputs, store them in the long-term state, retain them for as long as necessary, and extract the information as needed (Géron, 2022).

3.2 Multivariate LSTM Model

Feature engineering plays a fundamental role in the context of machine learning workflows and involves the transformation of data, aiming to create an optimal format that effectively captures the underlying problem that a machine learning algorithm seeks to address (Wang et al., 2022). The input for the machine learning model should be prepared in such a way, that it can better recognize the underlying data relations (Ozdemir, 2022). In the context of time series, Ozdemir (2022) distinguishes between date and time features, lag features, which include earlier values (see Subsection 3.2.3 Lags), rolling window features, which include earlier time periods (see Subsection 3.2.5 Previous consumption on the same weekday), and domain-specific features. In the energy consumption domain, it is important to consider various factors that influence electricity consumption, such as weather conditions, social norms, and periodic patterns throughout the day, week, or year (Weber et al., 2021).
In the following subsections, the rationale behind the feature selection and their properties are discussed in detail. The following features are considered:

- Weekday
- Calendar Week
- Lags
- Outdoor Temperature
- Previous consumption on the same weekday
- Similar user consumption
- Household Occupancy

3.2.1 Weekday

Considering the variations in energy behavior across households during weekdays and weekends, it is reasonable to include a weekday feature (Brownlee, 2018). Weekdays (Mo-Fr) were transformed to numeric values (1-5) while the weekends and holidays were marked as 0. A transformation into categorical variables was refrained from, since the size of the dataset used in this study is insufficient. Furthermore, it was

decided not to number the days of the week because otherwise the model would make the assumption that Sunday (in that case, 7) and Monday (1) are far apart, while with the chosen transformation the relationship only between Friday and Saturday is not apparent, which was considered to be less important. Alternatively, a simple boolean variable describing whether a day is a holiday or not can be considered.

3.2.2 Calendar Week

Since the sequences used as input only covered 3 weeks (see 5.1 Model, Training and Validation), it was decided to extract another time variable, the calendar week. This feature was implemented to ensure that the temporal aspect, respectively the position of the sequence in a year, could be captured in the sequence. For this purpose, the corresponding calendar week (1-52/53) of the sequences was added as a numeric value.

3.2.3 Lags

The use of time series data allows us to use past values as input features to determine future values (Ozdemir, 2022). In this study, energy consumption values in kWh of lag 1 were constructed and added to the sequence. This feature adds for each day of the sequences, with which the model is trained, the energy consumption value of the previous day. Past daily consumption values are already part of the sequence, but with this feature the aim is to strengthen the link between two consecutive days and thus, amplifying the weights of the energy consumption in t for the prediction in $t + 1$. For example, if a simple sequence of electricity consumption over three days is 2.0 kWh, 3 kWh and 4.0 kWh the corresponding Lag 1 sequence would be 1.0 kWh, 2.0 kWh, 3 kWh where 1.0 kWh corresponds to the consumption on a day prior to the beginning to the main sequence.

3.2.4 Outdoor Temperature

As described in a review about data-driven techniques for modeling and forecasting building energy consumption by Bourdeau et al. (2019), the impact of outdoor air temperature on building energy behavior is well-established. In order to construct

this feature, the average daily temperature of *St. Imier, Switzerland* in degrees Celsius was obtained using the Meteostat Python library (Meteostat, 2022), an open-access weather database. The temperature data was then shifted back by one time step so that the input sequence contains the temperature in $t + 1$ to predict the energy consumption in $t + 1$. The temperature values were standardized between -1 and 1 using the MinMaxScaler from the scikit-learn library (scikit-learn, 2023).
The subsequent equation presents the formula used for standardization:

$$T_{\text{scaled}} = \frac{T - T_{\text{min}}}{T_{\text{max}} - T_{\text{min}}} \times (\text{max} - \text{min}) + \text{min} \tag{2}$$

where:

- T represents the temperature in degrees Celsius,
- T_{min} represents the minimum value of T in the dataset,
- T_{max} represents the maximum value of T in the dataset,
- min represents the desired minimum value for the scaled data range (-1)
- max represents the desired maximum value for the scaled data range (1)

This transformation is commonly used to reduce the impact of absolute scales on the learning process.

3.2.5 Previous consumption on the same weekday

The average consumption in kWh on the two previous same weekdays was included in the feature set. This feature also represents at the same time the prediction method used for the benchmarking baseline (see 3.3 Benchmarking baseline). This type of feature is generally referred to as a rolling window feature, where past data points are combined into one feature (Ozdemir, 2022). The prediction of one day can be improved by utilizing this feature, as it captures existing patterns on specific weekdays.

3.2.6 Similar household consumption

The average load profile in kWh of the three most similar households were extracted. To find a similar household h_s for a unique household h_u, the following similarity metric using the euclidean distance was developed:

$$\text{Similarity}_{h_u,h_s} = \sqrt{(\bar{h_u} - \bar{h_s})^2 + (\bar{h}_{u,\text{morning}} - \bar{h}_{s,\text{morning}})^2 + (\bar{h}_{u,\text{evening}} - \bar{h}_{s,\text{evening}})^2} \quad (3)$$

where:

- $\bar{h_u}$ represents the average consumption of household h_u.
- $\bar{h_s}$ represents the average consumption of household h_s.
- $\bar{h}_{u,\text{morning}}$ represents the average morning consumption of household h_u during 6:00 am to 8:00 am.
- $\bar{h}_{s,\text{morning}}$ represents the average morning consumption of household h_s during 6:00 am to 8:00 am.
- $\bar{h}_{u,\text{evening}}$ represents the average evening consumption of household h_u during 7:00 pm to 9:00 pm.
- $\bar{h}_{s,\text{evening}}$ represents the average evening consumption of household h_s during 7:00 pm to 9:00 pm.

By considering the average consumption values for the whole day as well as the average consumption values during certain periods (morning and evening), different aspects of the consumption patterns were captured that could contribute to the similarity or dissimilarity between the households. The average load profile of similar users was then shifted back by one time step. This means that, for each day in the input sequence, the average consumption of similar users on the previous day is added. The purpose of this approach is to incorporate the load profile of similar users in $t+1$ into the prediction of energy consumption for the unique household in $t+1$. The input sequence then contains the load profile of similar users in $t+1$ to predict the energy consumption of the unique household in $t+1$.

3.2.7 Household Occupancy

The occupancy of the household is another known energy driver (Bourdeau et al., 2019). Since the dataset lacked specfic information regarding occupancy, a classification formula using a threshold value was developed after analyzing the visualized load profiles to allow for a binary coded feature:

$$\text{is\_home}_{d,i} = \begin{cases} 1, & \text{if daily consumption}_{d,i} > 0.6 \times D_{m,i} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where:

- $\text{is\_home}_{d,i}$: Binary variable indicating if the household for individual i is occupied on day d.
- $\text{daily consumption}_{d,i}$: Energy consumption for individual i on day d.
- $D_{m,i}$: Average daily consumption for individual i in month m.

If the daily consumption exceeds 0.6 times the average daily consumption within the given month, it indicates presence of the household members.

3.3 Benchmarking baseline

To obtain a benchmark for the performance of the LSTM models, different baselines (*historical averages*, *linear interpolation*, *forward fill imputation*) were compared and the best performing one, historical averages, was chosen. Specifically, this baseline relies on a straightforward assumption, that the prediction of today's energy consumption is equal to the average of the past two energy consumption values observed on the same weekday. Similar assumptions have been made in previous studies. Lavin and Klabjan (2015) used the same day of the week a year or two years earlier. Yilmaz et al. (2019) replaced missing values by values obtained at the same time one week later whereas Perret et al. (2018) selected three periods with identical weekday and time of day, taking into account the occupancy of the household as well.

4 Data

As the nature of the data can affect a model's performance, careful data cleaning needs to be carried out (Zheng, 2015). Adequate pre-processing, including the treatment of missing data, is essential for effective model training (Bourdeau et al., 2019). To prepare the load curves for the training and evaluation of daily intervals, data analysis and pre-processing of the datasets with the following time-intervals was applied:

- Data Analysis of 15-Minutes Intervals
- Data Analysis of Hourly Intervals
- Data Analysis of Daily Intervals
- Dataset for Training and Validation

An explanation of the performed steps is presented in the subsections below. The dataset used in this study was provided by a regional energy distribution industrial partner. The consumption values were already transformed from absolute values, representing the total amount of energy used at a given date and time to a format indicating consumption per time intervals.

4.1 Data Analysis of 15-Minutes Intervals

The loaded parquet file consists of approximately 80 million entries and three columns that indicate an anonymized household ID, with time and the energy consumption. Entries cover a 3-year period, but data are not available for all 15-min intervals of each household. Analysis of the amount of unique values per household ID and date column revealed that there are 1000 unique IDs and 105'216 unique date values, indicating that there is at least one entry every 15 minutes over the three years. In order to handle implausible data on 15-min intervals, a condition as outlined by Perret et al. (2018) was applied in order to mitigate the impact on the research outcomes in the subsequent steps.

The condition ensures that the valid energy consumption is strictly non-zero, even for the 15-minute interval:

$$E_{i,t} > 0 \tag{5}$$

where $E_{i,t}$ is the energy consumption in kW in a time interval t for individual i.

19.90% of the dataset contained values of 0 or negative, which were subsequently identified and designated as missing observations.

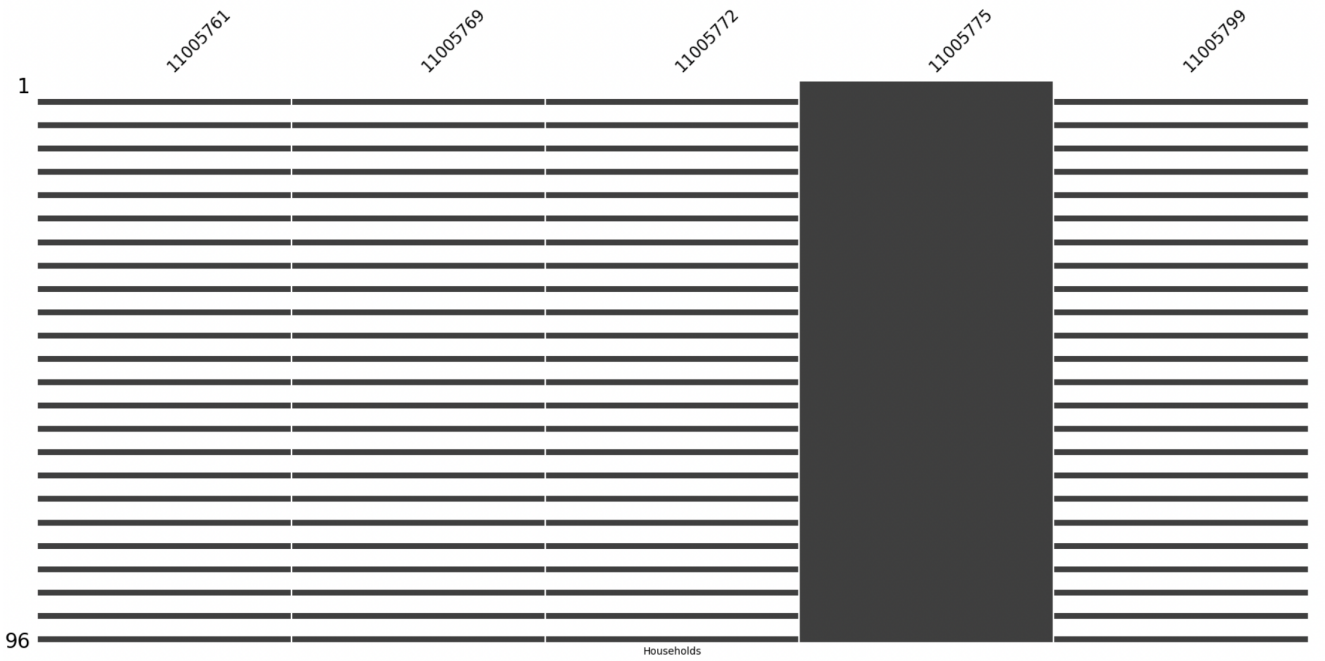
Based on the analysis of the remaining dataset, it was found that 9.86% of the rows are duplicated with the same date and ID value. Since no information was available on how to extract meaningful load curves from the duplicates, households that had duplicate content were omitted for further analysis. When visualizing the location of the missing values, it was found that for considerable number of households a consistent pattern was observed where only a single 15-minute entry was present within each hour, indicating that it likely represented the hourly value, while the remaining three 15-minute intervals were missing. Figure 5 illustrates this property showing the distribution of missing values for 5 households in a period of one day (96 15-min intervals). The x-axis represents the different households, whereas the y-axis represents the time intervals.

This observation could possibly indicate that these data points originate from older smart meters that only provides hourly read-outs. When performing the transformation of values from 15-minute intervals to 1-hour intervals, considering the presence of older smart meters, the following generalized restrictions were proposed for a given hour:

$$\text{hourly\_data}_i = \begin{cases} \text{sum}(\text{data\_intervals}_i) & \text{if all four intervals are available,} \\ \text{data\_intervals}_i[0] & \text{if only the first interval is available.} \end{cases}$$

If all four intervals are available for a given hour, the given hour is defined by summing up the four intervals. If only one interval is available for a given hour and it is the first interval of the given hour the value of this interval is interpreted

Figure 5 — Missing value locations for five selected households



Notes: The first three households, as well as the last household, exhibit this pattern, with the fourth household displaying values for each 15-minute interval. In the figure, the missing values are depicted as white regions. Own work.

as an hourly value. In any other case, the hourly data is considered as missing for the purposes of this study. By imposing this restriction, data were not summed when values were only available at other intervals, which would indicate true 15-minute intervals rather than hourly data.

4.2 Data Analysis of Hourly Intervals

The completion of the previous steps has resulted in the availability of a dataset comprising hourly intervals for a total of 914 users. This dataset has a missing data rate of 23.87%, indicating that approximately one-fourth of the hourly intervals are unavailable. This study makes use of an additional condition proposed by Perret et al. (2018) to eliminate implausible data on an hourly basis. According to this condition, during a one-hour interval, the average energy consumption should not be greater than 8kW.

$$P_{i,t} = \frac{E_{i,t}}{\Delta t} < 8\,\mathrm{kW} \quad \text{if } \Delta t = 1\,\mathrm{h} \tag{6}$$

where $P_{i,t}$ is the average energy consumption in kW in a time interval t for individual i.

Upon applying this restriction, it was discovered that the concerning households exhibited abnormally high outliers in their consumption data, frequently observed during February 2020. A possible cause of these high values can be the improper processing of the log data from the smart meters. A smart meter usually records the absolute number of electricity consumption. One plausible scenario is the following: the smart meter fails to send the absolute number for several days and then resumes sending data. By taking the difference from the latest reading and falsely attributing it to a single 15-minute interval, outliers can be created. Instead, the accumulated consumption should be spread out over the entire period of failure.

As a next pre-processing step and in accordance with the methodology outlined by Perret et al. (2018) a valid data rate for each load curve is defined as:

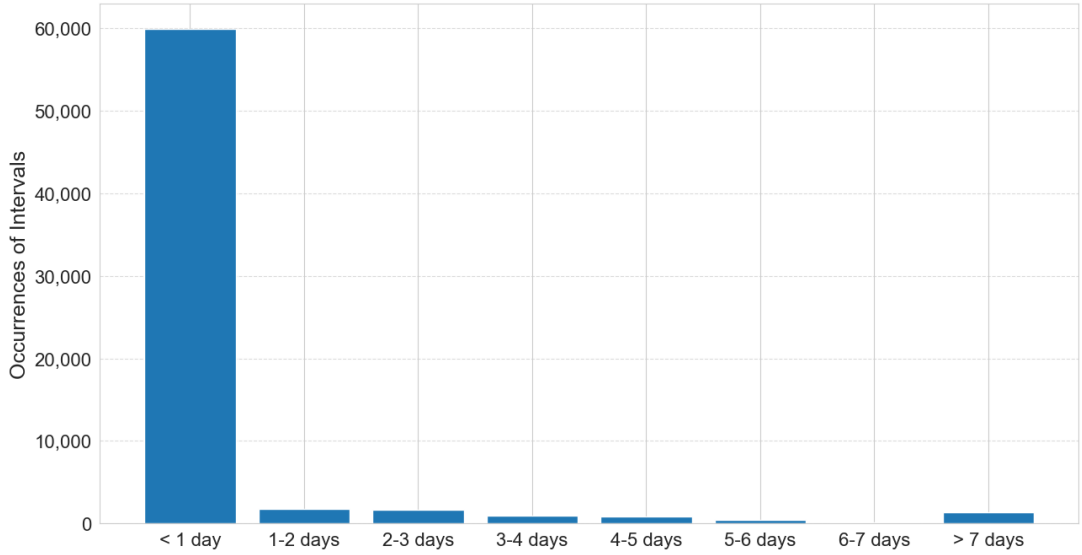
$$\frac{n_i}{T} \geq 0.7 \tag{7}$$

where n_i represents the number of available intervals in the load curve of a household i and T stands for the total number of hourly intervals in a three-year period.

While Perret et al. (2018) applied a valid data rate of 50%, in the case of this study a higher threshold of 70% is implemented to minimize the potential for biased model training. As a result, the amount of households in the dataset dropped from 914 to 691. The dataset still exhibited a missing value rate of 12.75 %.

In a next step, the dataset was examined to determine the common duration of missing values, in order to identify specific intervals that occur frequently and are likely to be encountered in the field experiment. Figure 6 shows the distribution of the occurrences of hourly missing values among the possible intervals. The x-axis segregates the intervals into seven categories, while the y-axis represents the frequency of these intervals without taking the length of the intervals into account. For example, an interval of 1 week of missing values is weighted the same as an interval of 0.5 day.

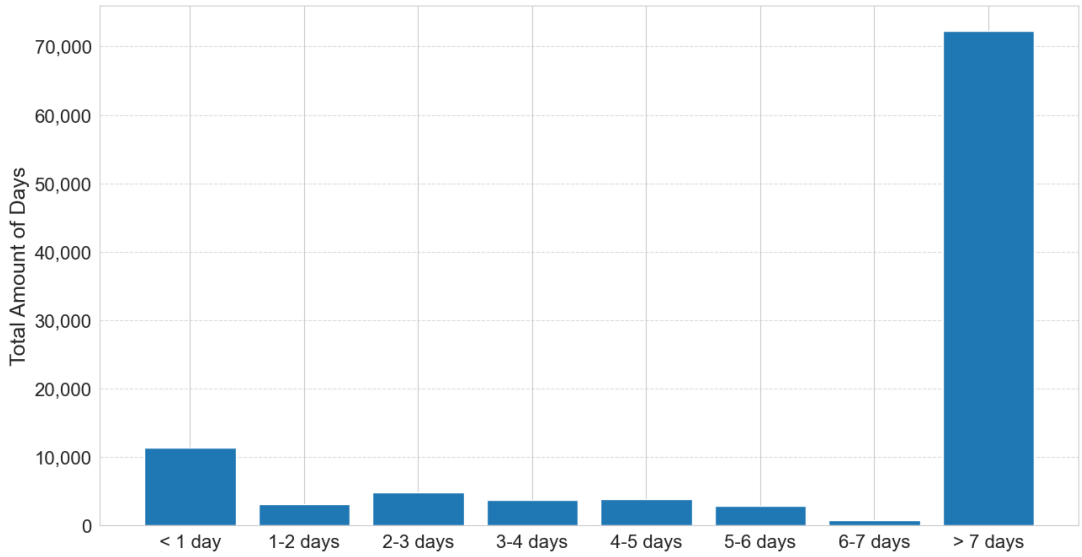
Figure 6 — Occurrences of Missing Data Intervals



Notes: Own work.

Based on the occurrence, there are mainly missing data in intervals of less than one day and of more than seven days. When adjusting for the duration of each respective interval, it can be observed in Figure 7, that the distribution of the total amount of missing values is mainly attributable to households with very long intervals and missing values within a daily period.

Figure 7 — Total Amount of Days for each Missing Data Interval



Notes: Own work.

This finding is especially of interest when access to raw smart meter data is available. Missing intervals within a day can then be treated by resampling the raw smart meter rather than constructing the daily records by summing the hourly records (current implementation). Smart meters usually send the value of absolute energy consumption at regular intervals. To calculate the consumption volume between two data points, the first point is subtracted from the last for each interval. By taking the difference between last seen and first seen data point, the missing volume consumed can be accurately inferred so that all missing values which are less than 24 hours can be treated. As the raw smart meter data were not provided in this study's time-frame, another process step was applied to reduce short missing intervals. Typically, missing intervals shorter than two hours are filled in using linear interpolation (Peppanen et al., 2016). In this particular case, missing intervals up to four hours were filled in to strike a balance between the need for data completeness and the understanding that the precision of hourly data is not crucial for the main objective of this study, which is focused on daily intervals. When resampling from hourly to daily intervals, only days with 24 hours of available data were summed. If less than 24 hours of data were available for a day, the daily value for that day was considered missing.

4.3 Data Analysis of Daily Intervals

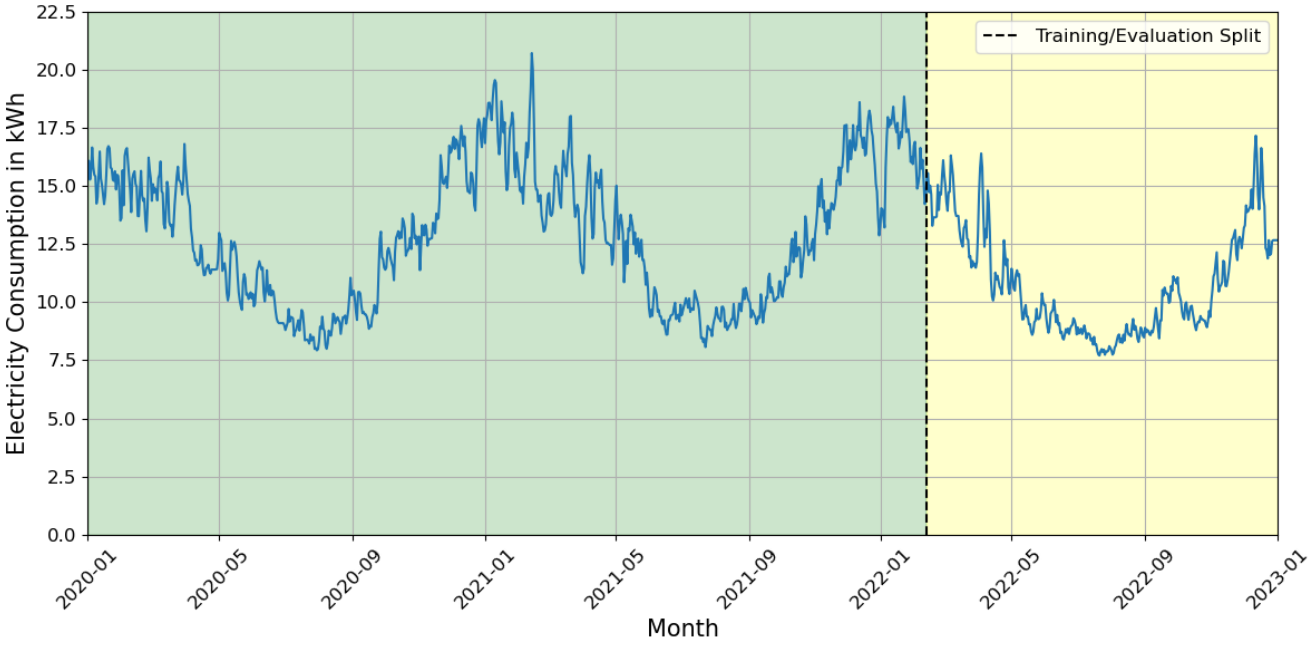
Due to the limited size of the dataset and a frequent presence of missing values, it is not possible to restrict the study only to the samples of users without missing values. Therefore, potential missing values are addressed as follows for training purposes. If there was only one day missing the average of the consumption from the previous day and the following day was taken in order to reduce the amount of missing values. To identify outliers within the daily intervals, another adjustment was made. Any values exceeding four times the average consumption were flagged as missing values. This determination is based on a visual analysis of the individual load curves. This approach was introduced to account for extreme deviations from expected consumption patterns that might indicate measurement errors or unusual circumstances. Also, daily consumption values that were less than 0.5 kWh were marked as missing. This procedure is based on Yilmaz et al. (2019)'s paper, which describes daily values below 0.5 kWh as implausibly low. In order to reduce potential biases in model training and evaluation, only households with no missing intervals greater than 21 days were selected resulting in a training and validation dataset of 212 households. For the remaining 4.31% of missing values, the method of Last Observation Carried Forward (LOCF) was applied to have full sequences of data in order to train and evaluate the model.

4.4 Dataset for Training and Validation

The final dataset contains energy consumption loads of 212 households during a period of three years in daily intervals. Figure 8 illustrates the load curve of the average daily electricity consumption across all households. Significant fluctuations can be observed over the three years, mainly due to seasonal reasons and social norms (e.g. New Year holidays). The average electricity consumption per household is 4493 kWh/year, which is in the same range as other available data in Switzerland. For example, Axpo (n.d.), a Swiss-based energy company, estimates for an average

four-person household between 4500 and 5000 kilowatt hours of electricity per year (including electric water heating).

Figure 8 — Average Daily Energy Consumption



Notes: Own work.

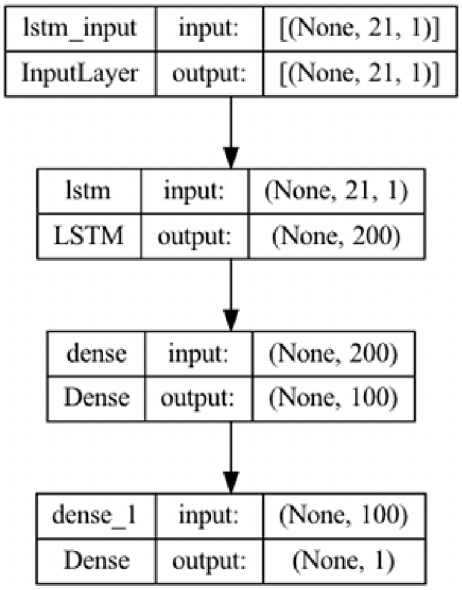
5 LSTM Setup

5.1 Model, Training and Validation

For the training and validation of the model, each individual household is considered separately in order to take into account household-specific consumption. As can be seen in Figure 8, the load curves are divided into training and validation using a common 70%/30% split. Thus, the training set for each household consists of data spanning from January 01, 2020 to February 12, 2022, while the period from February 13, 2022 to December 31, 2022 is available for validation of the trained model. This split is a crucial step to mitigate overfitting by allowing the trained model to be evaluated using unseen data, thereby enhancing its generalization capabilities (Géron, 2022). The model configuration was adopted from Brownlee (2017)'s book *"Predict the Future with MLPs, CNNs and LSTMs in Python"* in which Chapter 20 discusses the development of LSTMs for energy consumption prediction. The choice of sequence length influences the acquisition of high-level features that can be learned in sequential data (Yang et al., 2019). In this work a sequence length of 3 weeks was chosen based on empirical observations of training performance. It was motivated by the need of capturing the weekly cycle and consumption trends while considering the limited volume of data. An LSTM model requires input data in a three-dimensional format [samples, steps, features]. *Samples* are the number of possible sequences in a defined time period, which are used as input to train the model. In this study, the time period considered is between January 01, 2020 and February 12, 2022 and sequences of 21 days are used. The output is defined as the expected daily electricity consumption on the day of prediction. Thus, the first input includes energy consumption values from January 01, 2020 to January 21, 2020 and the output includes the energy value on January 22. The sequences and output values move along one time step (one day) until the end date is reached (Brownlee, 2018). *Time steps* is the length of the sequence, in the case of this study, 21 (days). *Features* are the provided variables to the model inputs. The model

introduced in 3.1 Univariate LSTM Model utilizes a single feature, specifically the past consumption data, whereas the model discussed in 3.2 Multivariate LSTM Model employs multiple features. As the goal is to predict electricity consumption of a given household on the following day (prediction of 1 day), the output value is equal to the electricity consumption on that day. The developed LSTM model consists of a hidden layer with 200 neurons, activated with the ReLU activation function. Following the LSTM layer, a fully linked dense layer with 100 nodes is used that aims to interpret the features learned from the LSTM layer. The output layer predicts a single numerical value corresponding to the day prediction. The overall architecture of the model is shown in Figure 9.

Figure 9 — LSTM Model Architecture



Notes: *None* represents the number of samples for the model training. For the LSTM Model in this study, there are samples with sequences of 21 days. Generated by the author using TensorFlow's plot_model function. Own work.

Epochs are the number of times the model is exposed to the whole training set. *Batch Size* stands for the the number of samples within an epoch after which the weights are updated (Brownlee, 2018). The training undergoes 70 epochs, using a batch size of 16. The selected values were chosen based on performance observations.

In a subsequent step, the trained LSTM model is then used for prediction using

the validation dataset where this study makes use of a walk-forward validation (Brownlee, 2018). In the set-up of this study, the first validation step uses an input sequence of 21 days from February 13 to March 4 to make a prediction about March 5. Walk-forward in this context means that the true value of March 5 is then provided to the model to make a prediction about March 6, and so on. These validation steps were performed until December 31, 2022 is reached. The model's performance is evaluated by calculating two different error metrics, Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). The two error metrics are presented in detail in Subsection 5.2 Error metrics. The performance of the two models is discussed in Chapter 6 Results.

5.2 Error metrics

The performance of a predictive LSTM model is assessed by the level of similarity between the predicted values and the ground truth. This similarity, respectively difference can be quantified through various methods (Banachewicz & Massaron, 2023). The first validation metric used in this study is the mean absolute error (MAE). It is defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y}_i - y_i| \tag{8}$$

where n stands for the number of predictions, $\widehat{y}_i$ represents the prediction, and y_i is the ground truth.

The MAE calculates the average of the absolute differences between actual and predicted values and is thus not sensitive to large prediction errors (Banachewicz & Massaron, 2023).

When the objective is to understand the relative magnitude of errors between a prediction and the ground truth percentage errors such as the Mean Absolute Percentage Error, MAPE for short, can be considered (Zheng, 2015).

MAPE is based on the ratio of the absolute error to the ground truth value:

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\widehat{y}_i - y_i}{y_i} \right| \times 100 \tag{9}$$

where n is the number of predictions, $\widehat{y}_i$ represents the predicted value, and y_i is the actual value.

6 Results

A fundamental contribution of this study is the development of a feature engineering and LSTM network processing infrastructure. The results presented in this chapter were obtained using the dataset described in Subsection 4.4 Dataset for Training and Validation. The technological solution developed is designed to be easily reused with new datasets that are expected to contain better quality data or additional relevant features. In such a case, it is expected that the performance shown here will improve accordingly.

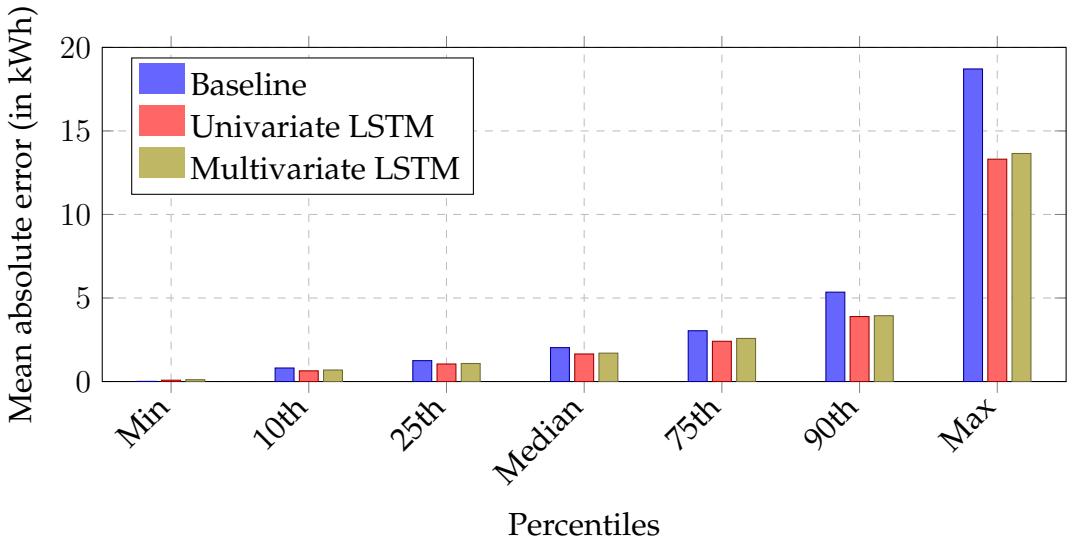
6.1 Performance overview

The following tables and figures provide an evaluation of the three main methods using the discussed error metrics in Subsection 5.2 Error metrics. While Table 1 employs the Mean Absolute Error (MAE), Table 2 uses the Mean Absolute Percentage Error (MAPE) for assessment. Values are visually represented in Figure 10 (MAE) and Figure 11 (MAPE). For the multivariate LSTM model, multiple configurations were evaluated and the one leading to the highest performance is selected in this chapter. This configuration incorporates the features *Weekday*, *Temperature*, *Previous consumption on the same weekday*, *Similar user consumption*, and *Household Occupancy*.

Table 1 — Mean absolute error (MAE)

Percentiles	Mean absolute error (in kWh)		
	Baseline	Univariate LSTM	Multivariate LSTM
Min	0.01	0.08	0.11
10th Percentile	0.81	0.64	0.69
25th Percentile	1.25	1.05	1.08
Median	2.03	1.65	1.70
75th Percentile	3.04	2.41	2.58
90th Percentile	5.35	3.89	3.94
Max	18.71	13.31	13.65
n	212	212	212

Figure 10 — Comparison of performance (MAE)



Notes: Own work.

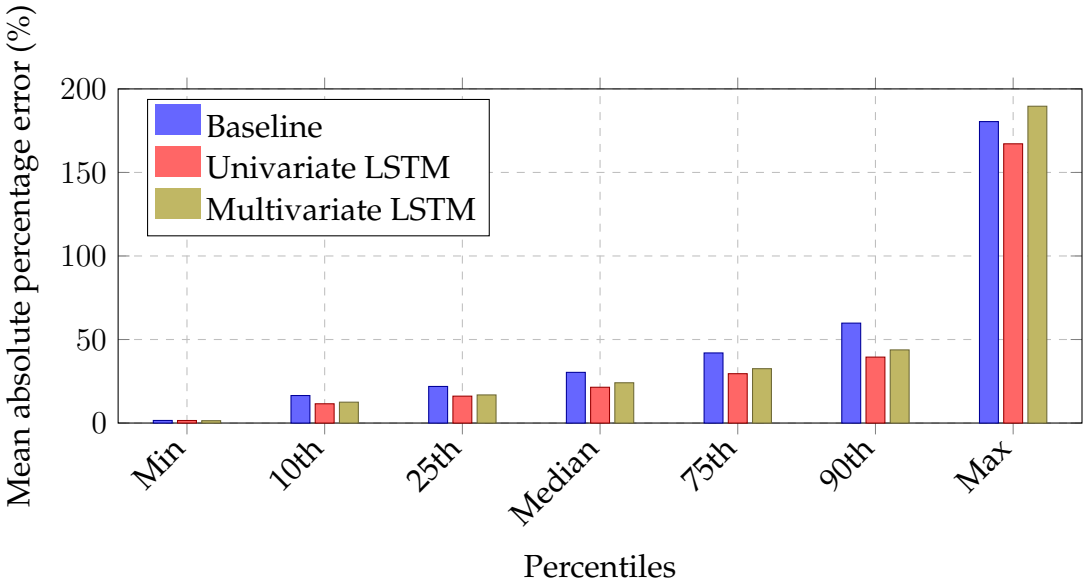
Table 1 and Figure 10 presents the distribution of the Mean Absolute Error among the 212 households. It can be observed that the baseline model obtains the highest MAE values over all percentiles, indicating the least accurate predictions. Both, the univariate LSTM and multivariate LSTM models lead to better predictions than the baseline model, although the multivariate LSTM model did not outperform the univariate LSTM model. Given an average daily consumption of 12.23 kWh in the dataset, the univariate LSTM model achieved a MAE of 2.41 kWh at the 75% percentile. In comparison, the multivariate LSTM model demonstrated a slightly higher MAE of

2.58 kWh at the same percentile. The univariate LSTM achieved a median absolute difference between the predicted values and the actual values of 1.65 kWh whereas the multivariate LSTM model recorded a median absolute difference of 1.70 kWh.

Table 2 — Mean absolute percentage error (MAPE)

Percentiles	Mean absolute percentage error		
	Baseline	Univariate LSTM	Multivariate LSTM
Min	1.55%	1.52%	1.41%
10th Percentile	16.47%	11.53%	12.49%
25th Percentile	21.90%	16.10%	16.83%
Median	30.34%	21.42%	24.11%
75th Percentile	41.96%	29.54%	32.54%
90th Percentile	59.81%	39.47%	43.78%
Max	180.44%	167.14%	189.63%
n	212	212	212

Figure 11 — Comparison of performance (MAPE)



Notes: Own work.

Table 2 and Figure 11 show the distribution of the MAPE evaluation for the 212 households. It is important to note that MAPE is a scale dependent metric, which means that the interpretation of the error varies depending on the size of the target variable. Again, the two LSTM models outperform the base model in almost all

percentiles and show more accurate predictions. This result confirms the assumption that using machine learning models in the context of energy consumption leads to improvements in quality of predictions. The median MAPE for the univariate LSTM model is 21.41%, which means that on average over all 212 households, the predictions deviate by 21.41% from the ground truth. This performance exceeds significantly the median MAPE of 30.34% established by the benchmark (see Subsection 3.3 Benchmarking baseline). Between the two LSTM models, the univariate LSTM model consistently outperforms the multivariate LSTM model over all percentiles. Therefore, in the view of the observed performance, machine learning solutions developed in this work are going to be used in INFINEED project for the treatment of missing values.

6.2 Performance based on load curve characteristics

In order to break down the performance even further, two energy consumption characteristics are developed to identify household types, where the LSTM model performs particularly well. For the first categorization, the total energy consumption for each household over the three years is investigated and the households are categorized into three branches, based on the 33rd, 66th, and 99th percentiles (low, medium, high). For the second category, the seasonality of each household is explored by fitting sinusoidal curves to their load curves in order to determine whether they exhibit seasonal patterns. These two categories allow for six different subgroups which are shown in Table 3 where values indicate the subgroup sample size n. Table 4 presents the average daily consumption in kWh for each subgroup.

Table 3 — Load curve subgroup size (n)

		Energy consumption volume		
		Low	Medium	High
Seasonality	Present	14	38	49
	Absent	55	32	24

Table 4 — Average daily consumption (kWh) for each subgroup

		Energy consumption volume		
		Low	Medium	High
Seasonality	Present	4.77	9.18	25.88
	Absent	4.13	8.29	18.36

The subsequent discussion focuses on the results obtained from the three models, considering the six subgroups. Important to note here is that the maximum values should be interpreted with caution. Maximum values are by definition at the extreme end of the distribution and can be influenced by outliers or rare events that may not be representative of the general trend (Banachewicz & Massaron, 2023). For example, a large difference between the 90th percentile and the maximum value, especially compared to the spread between other percentiles, may indicate that the maximum value is an outlier. As a result, the following discussion will largely refrain from discussing maximum values. The subgroups are denoted in the table as follows:

pSL: Present seasonality, low energy consumption.

pSM: Present seasonality, medium energy consumption.

pSH: Present seasonality, high energy consumption.

aSL: Absent seasonality, low energy consumption.

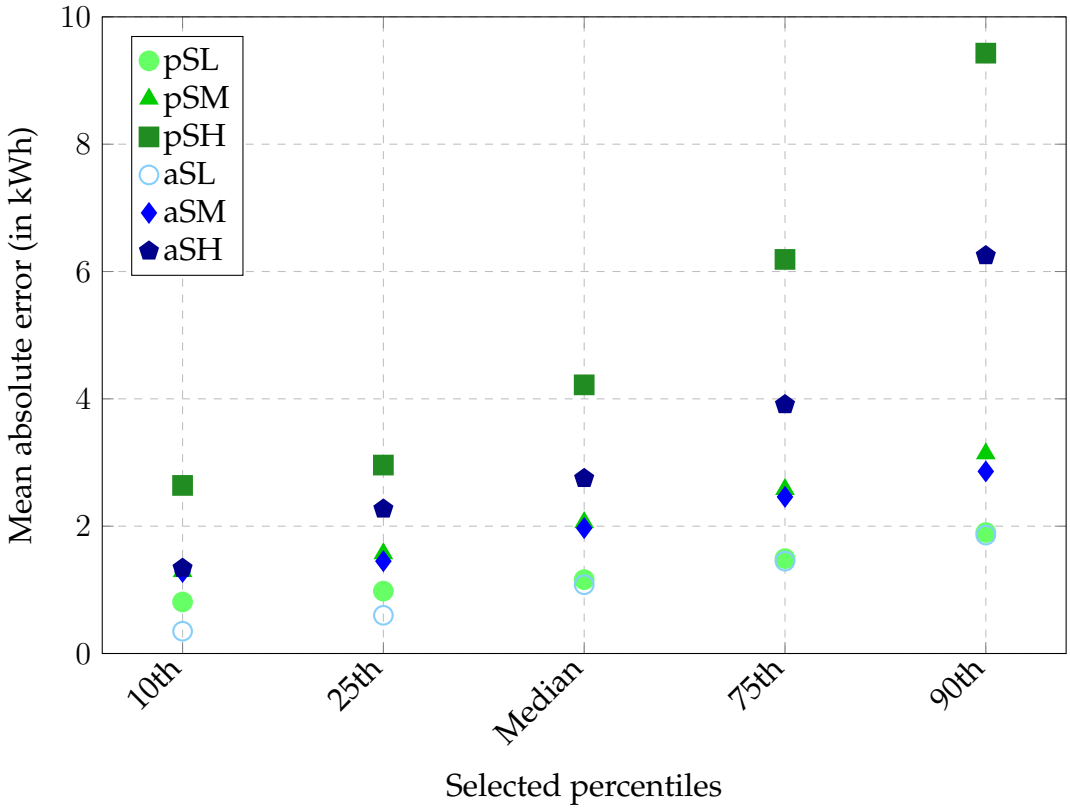
aSM: Absent seasonality, medium energy consumption.

aSH: Absent seasonality, high energy consumption.

6.2.1 *Benchmarking baseline*

Figure 12 respectively Table 5 show the MAE performance of the baseline model with respect to the 6 household groups. Subsequently, Figure 13 and Table 6 illustrate the corresponding MAPE values.

Figure 12 — Baseline Subgroup Comparison (MAE)



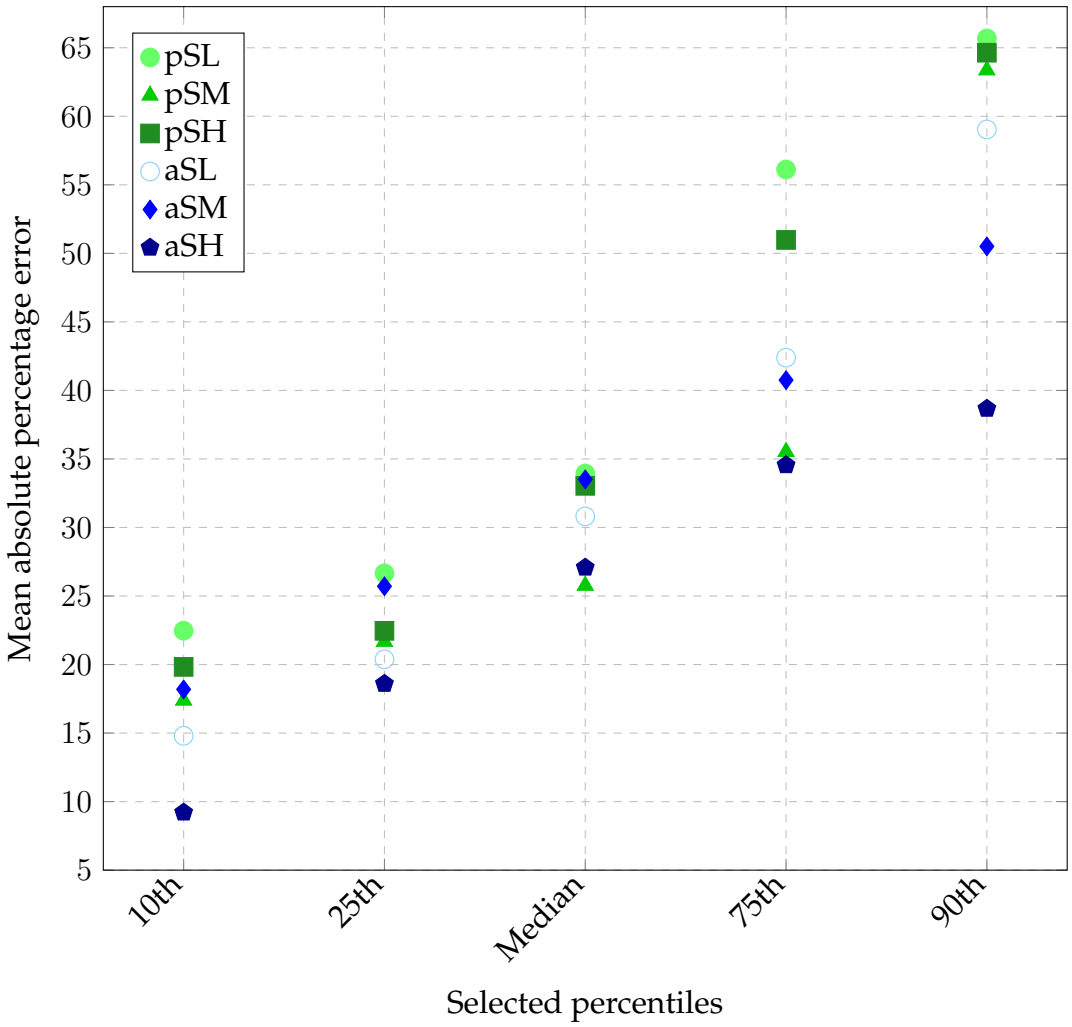
Notes: Own work.

Table 5 — Baseline Subgroup Comparison (MAE)

Percentiles	Mean absolute error (in kWh)					
	pSL	pSM	pSH	aSL	aSM	aSH
Min	0.62	0.89	2.07	0.01	1.01	0.78
10th Percentile	0.81	1.29	2.64	0.35	1.28	1.34
25th Percentile	0.98	1.57	2.96	0.60	1.45	2.27
Median	1.16	2.06	4.22	1.08	1.97	2.75
75th Percentile	1.49	2.58	6.19	1.45	2.46	3.91
90th Percentile	1.90	3.14	9.43	1.86	2.86	6.25
Max	2.94	3.60	18.71	2.77	6.58	7.33
n	14	38	49	55	32	24

Referring to Figure 12 and Table 5, the baseline model generally performs worse for the household groups with seasonality than their volume equivalent without seasonality. The median MAEs of the three subgroups exhibiting seasonal patters are consistently higher then the corresponding median MAEs of the subgroups without seasonality. This is quite intuitive, since for the baseline group the average of the energy consumption of the two previous same weekdays is taken as prediction, and in case of seasonality this prediction is always biased by the seasonal trend. Moreover, the two household groups characterized by a low energy volume exhibit the lowest MAE values across the percentiles.

Figure 13 — Baseline Subgroup Comparison (MAPE)



Notes: Own work.

Table 6 — Baseline Subgroup Comparison (MAPE)

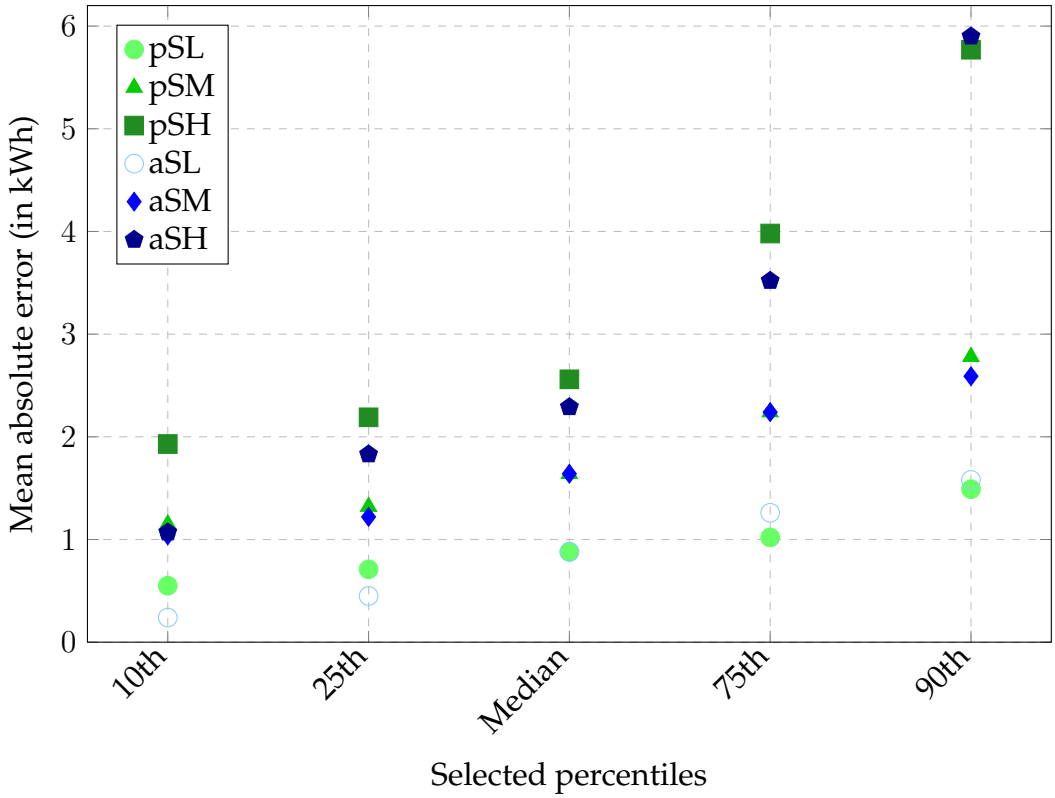
Percentiles	Mean absolute percentage error					
	pSL	pSM	pSH	aSL	aSM	aSH
Min	15.49%	10.96%	2.75%	1.94%	13.67%	1.55%
10th Percentile	22.47%	17.38%	19.83%	14.80%	18.19%	9.21%
25th Percentile	26.66%	21.67%	22.46%	20.38%	25.71%	18.60%
Median	33.94%	25.76%	33.04%	30.81%	33.51%	27.08%
75th Percentile	56.12%	35.52%	50.98%	42.39%	40.75%	34.55%
90th Percentile	65.68%	63.36%	64.64%	59.04%	50.51%	38.67%
Max	84.70%	88.97%	159.26%	106.73%	180.44%	67.80%
n	14	38	49	55	32	24

A similar picture, although not as clear, can be drawn by the MAPE values of the different subgroups (see Figure 13 and Table 6). Focusing on the 90th percentile, the MAPE of the three subgroups that exhibit seasonality are 65.68%, 63.36%, and 64.64%. In comparison, for the household groups without seasonality, MAPE values of 59.04%, 50.51%, and 38.67% are observed. The lowest MAPE values are achieved by the household group with absent seasonality and high energy consumption (aSH), which reaches a median MAPE of 27.08%.

6.2.2 *Univariate LSTM Model*

The performance of the univariate LSTM model, as presented in Figure 14 and Table 7 for MAE values, and Figure 15 and Table 8 for MAPE values, will be discussed subsequently.

Figure 14 — Univariate LSTM Subgroup Comparison (MAE)



Notes: Own work.

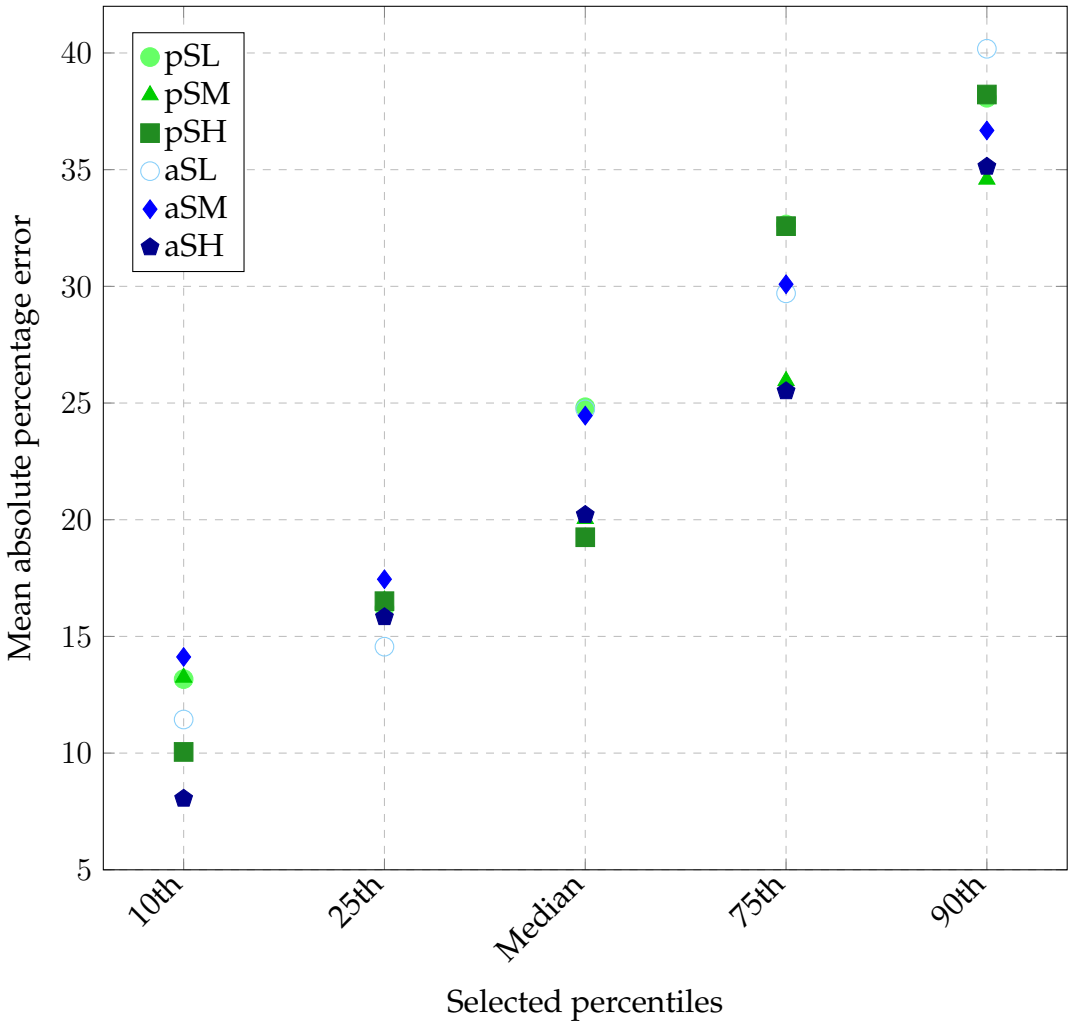
Table 7 — Univariate LSTM Subgroup Comparison (MAE)

Percentiles	Mean absolute error (in kWh)					
	pSL	pSM	pSH	aSL	aSM	aSH
Min	0.54	0.67	1.45	0.08	0.90	0.77
10th Percentile	0.55	1.15	1.93	0.24	1.04	1.07
25th Percentile	0.71	1.32	2.19	0.45	1.22	1.83
Median	0.88	1.64	2.56	0.88	1.64	2.29
75th Percentile	1.02	2.24	3.98	1.26	2.24	3.52
90th Percentile	1.49	2.78	5.77	1.58	2.59	5.90
Max	2.41	3.35	13.31	2.36	5.00	6.99
n	14	38	49	55	32	24

It can be observed, that there were particularly low values for the MAE of households with a low energy volume and no seasonality (see Subsection 6.2 Performance based on load curve characteristics) in their load curve, with an median MAE of 0.88 kWh, ranging from the 10th percentile at 0.24 kWh to the 90th percentile at 1.58 kWh. The group of households with low energy volume and seasonality also show low MAE values with a median MAE of 0.88 kWh, ranging from the 10th percentile at 0.54 kWh to the 90th percentile at 1.49 kWh. This confirms the expectation that households with low energy volumes also have lower absolute errors in their predictions.

The subgroup characterized by high energy volume and seasonality was identified as the household group for which the model's predictions were poor compared to other subgroups using the univariate LSTM model. In that case, the model demonstrated a performance characterized by a median MAE of 2.56 kWh, with the 10th percentile at 1.93 kWh and the 90th percentile reaching 5.77 kWh. This is understandable given our model did not considered users to be seasonal. The future model should consider the reasons for seasonal behaviour of certain users. For example, the model for describing seasonal users is likely to profit from additional customer information from the energy provider indicating the presence of a heatpump or electrical heating units. Also, seasonal and not seasonal users can be treated with completely independent models once more data is available in order to profit from underlying differences.

Figure 15 — Univariate LSTM Subgroup Comparison (MAPE)



Notes: Own work.

Table 8 — Univariate LSTM Subgroup Comparison (MAPE)

Percentiles	Mean absolute percentage error					
	pSL	pSM	pSH	aSL	aSM	aSH
Min	12.78%	8.45%	2.25%	5.68%	11.52%	1.52%
10th Percentile	13.17%	13.24%	10.05%	11.44%	14.12%	8.05%
25th Percentile	16.22%	16.54%	16.51%	14.56%	17.45%	15.83%
Median	24.82%	20.04%	19.25%	24.73%	24.46%	20.21%
75th Percentile	32.65%	25.95%	32.58%	29.70%	30.09%	25.51%
90th Percentile	38.08%	34.58%	38.22%	40.18%	36.68%	35.13%
Max	56.43%	167.14%	61.81%	57.17%	54.08%	46.42%
n	14	38	49	55	32	24

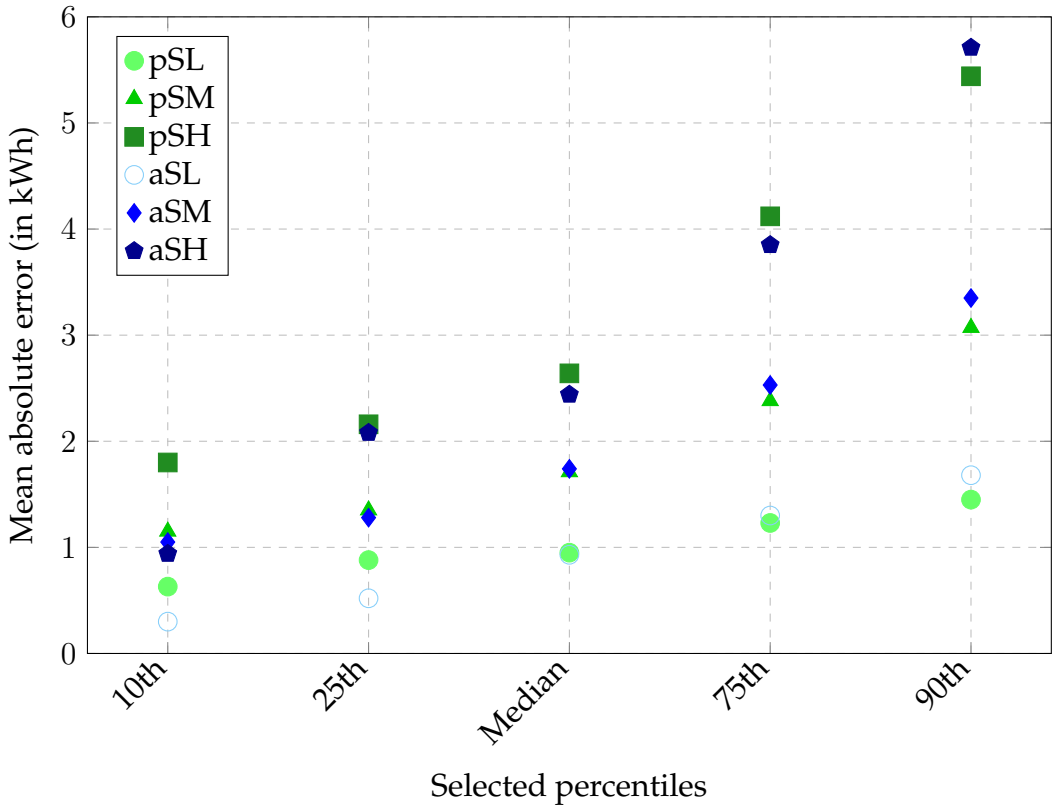
In terms of mean absolute percentage error (MAPE) as shown in Figure 15 and Table 8, the model demonstrated its best performance with the subgroup characterized by a high energy volume and no seasonality in their load curve. The median MAPE of this subgroup lies at 20.21%, with 8.05% at the 10th percentile and 35.13% at the 90th percentile. Meanwhile, the lowest performance was observed within the subgroup characterized by low energy volume and seasonality with an median MAPE of 24.82%, ranging from the 10th percentile at 13.17% to the 90th percentile at 38.08%. As mentioned earlier, households characterized by a seasonal pattern, could profit from a better performance if they are treated with a different model addressing their seasonal behaviour.

These results further strengthen the suitability of an LSTM model for the INFINEED project. Households with high energy consumption, where the model performs particularly well, are the focus of the project, as there is considerable potential for energy savings in these cases.

6.2.3 *Multivariate LSTM Model*

Similar trends to those discussed in 6.2.2 Univariate LSTM Model were observed in the analysis of the multivariate LSTM model as seen in Figure 16 and Table 9 for the MAE error metric and Figure 17 and Table 10 for MAPE.

Figure 16 — Multivariate LSTM Subgroup Comparison (MAE)



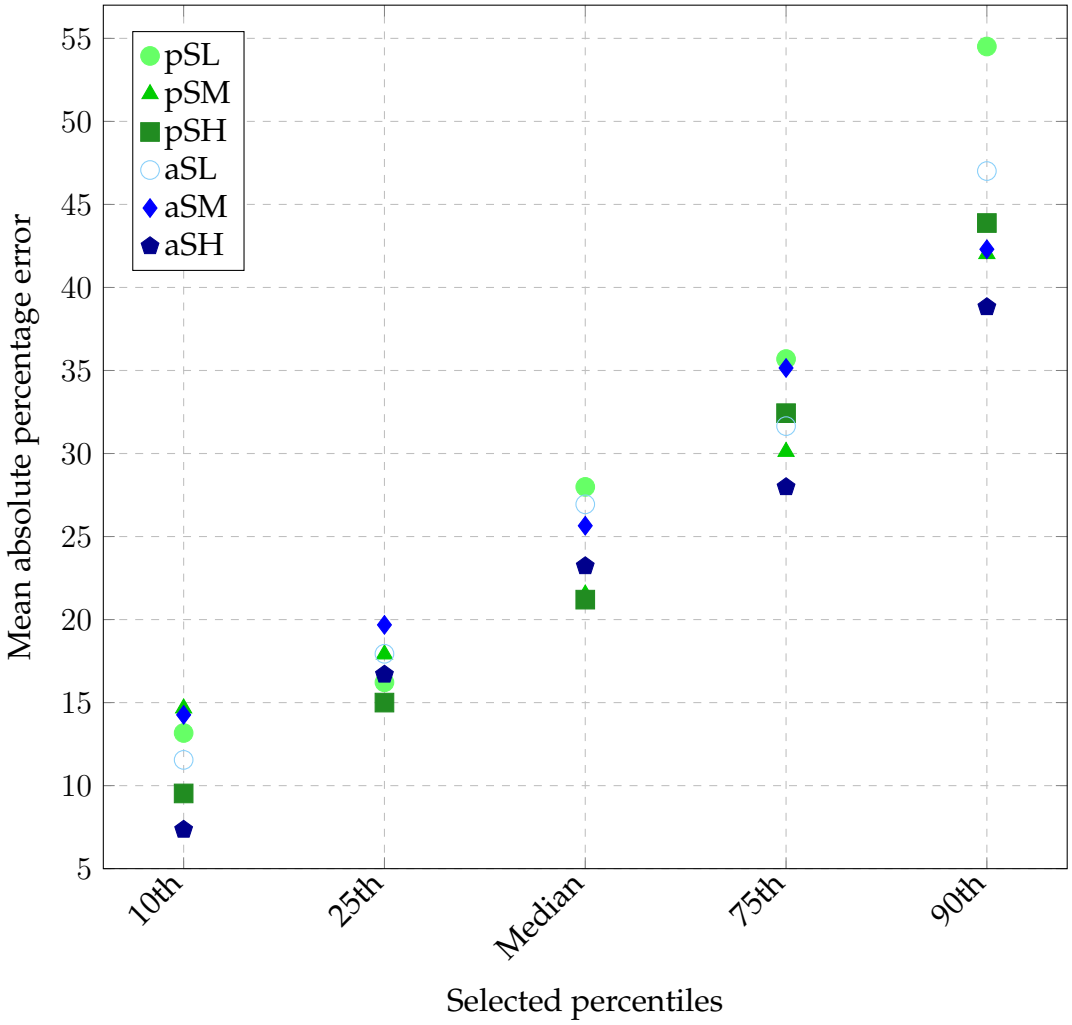
Notes: Own work.

Table 9 — Multivariate LSTM Subgroup Comparison (MAE)

Percentiles	Mean absolute error (in kWh)					
	pSL	pSM	pSH	aSL	aSM	aSH
Min	0.55	0.84	1.33	0.11	0.88	0.72
10th Percentile	0.63	1.15	1.80	0.30	1.05	0.94
25th Percentile	0.88	1.35	2.16	0.52	1.28	2.08
Median	0.95	1.71	2.64	0.93	1.74	2.44
75th Percentile	1.23	2.38	4.12	1.30	2.53	3.85
90th Percentile	1.45	3.07	5.44	1.68	3.35	5.71
Max	2.81	3.54	13.65	2.30	5.10	7.66
n	14	38	49	55	32	24

With the subgroup of households exhibiting a low energy volume and no seasonality, the best performance was achieved showing a median MAE of 0.93 kWh with the 10th percentile at 0.30 kWh and the 90th percentile at 1.68 kWh. To provide context, the average daily consumption of this subgroup is 4.13 kWh. When considering the lowest performance, both the univariate and multivariate models showed similar results, especially in the subgroup of households characterized by high volume and seasonal variations in their load behavior. In this subgroup, the multivariate model achieved an median MAE of 2.64 kWh, with the 10th percentile at 1.80 kWh and the 90th percentile at 5.44 kWh.

Figure 17 — Multivariate LSTM Subgroup Comparison (MAPE)



Notes: Own work.

Table 10 — Multivariate LSTM Subgroup Comparison (MAPE)

Percentiles	Mean absolute percentage error					
	pSL	pSM	pSH	aSL	aSM	aSH
Min	12.78%	12.48%	3.26%	7.19%	10.77%	1.41%
10th Percentile	13.17%	14.66%	9.53%	11.55%	14.26%	7.35%
25th Percentile	16.22%	17.91%	15.00%	17.94%	19.68%	16.69%
Median	27.99%	21.52%	21.20%	26.94%	25.65%	23.22%
75th Percentile	35.69%	30.11%	32.43%	31.65%	35.15%	27.98%
90th Percentile	54.51%	42.02%	43.88%	47.01%	42.30%	38.82%
Max	90.37%	189.63%	103.91%	83.90%	58.38%	59.89%
n	14	38	49	55	32	24

As for the MAPE, the best performance was achieved by the subgroup of households with no seasonality in their load curves and high energy volume. The median MAPE for this subgroup stands at 23.33%, with the 10th percentile at 7.35% and the 90th percentile at 38.82%. Using the households within the subgroup characterized by low energy volume and seasonality in their load curve, led to the worst performance with a median MAPE at 27.99% , with the 10th percentile at 13.17% and the 90th percentile at 54.51%. These results are similar to the performance of the household groups in the univariate model.

6.2.4 Alternative models

In Table 11, the performance of all executed combinations of features are listed. As mentioned in Chapter 6, the LSTM model, which simply contained past values over a period of 21 days as sequences, could not be outperformed.

Table 11 — Comparison of performance

Distribution	Mean absolute percentage error			
	(1)	(2)	(3)	(4)
Min	1.67%	1.64%	1.41%	1.60%
10th Percentile	13.01%	13.67%	12.49%	12.29%
25th Percentile	17.31%	17.94%	16.83%	17.25%
Median	23.72%	24.46%	24.11%	23.78%
75th Percentile	32.79%	32.36%	32.54%	33.02%
90th Percentile	43.76%	43.67%	43.78%	43.72%
Max	313.86%	186.40%	189.63%	190.95%
n	212	212	212	212
3.3.1 Weekday	x	x	x	x
3.3.2 Calendar Week	x	x	-	-
3.3.3 Lags	x	-	-	-
3.3.4 Temperature	x	x	x	-
3.3.5 Previous same-weekday consumption	x	x	x	x
3.3.6 Similar user consumption	x	x	x	x
3.3.7 Occupancy	x	x	x	-

7 Limitations and Future Work

A significant constraint of this study was data access, as the dataset obtained during the period of this study was limiting due to its quality (see Subsection 4.2 Data Analysis of Hourly Intervals), but also scope both in terms of volume and content. The predictive power of the model depends strongly on the quality of the available data. In this case, there was only access granted to a processed dataset of anonymised energy consumptions without any socio-economical attributes and it already contained many missing values. This led to the exclusion of a significant portion of the data for further analysis. Moreover, the remaining missing values had to be treated by imputation techniques, which in turn may potentially affect the results obtained. Therefore, this is considered on of the limitations of this study. A reevaluation of the presented models, once a better dataset becomes available, would remove a significant portion of missing value intervals (see Section 4.2 Data Analysis of Hourly Intervals) and therefore improve both training performance and result validity in the future.

It was observed that the multivariate models do not perform better than univariate model. This can be explained by several reasons. One of these possible explanation is the selection of the temperature of *St. Imier, Switzerland* as the outside temperature. It is evident that this specific temperature does not contain relevant information for the energy consumption curve of all households. Improvements can be achieved by generalizing, e.g. by taking the average cantonal temperature. Moreover, household specific location temperature would provide even more information about energy consumption. For this information, however, one is dependent on whether it is released by the electricity provider. Moreover, the variable "is_home" could be defined more precisely based on hourly data instead of being derived from daily consumption. Furthermore, features of seasonal users could be transformed using sine-cosine transformation which preserves the cyclical nature of the data (Chakraborty & Elzarka, 2019). In addition, the dataset contained only load curves without additional information such as the location, building size or information

about the electrical equipment (air conditioning and heating systems). The presence of such data could help to further narrow down the energy consumption predictions and thus improve the prediction accuracy. Moreover, another notable constrain was the limited computing power, which plays a crucial role in tasks such as hyper-parameter tuning, multistep predictions or the evaluation of machine learning models.

Building on this study, future work should include the mentioned multistep predictions such as predicting one or two weeks into the future to evaluate the performance of the LSTM model under such conditions. The analysis of Figure 7 has shown that missing values intervals over 7 days occur to a considerable extent. On top of that, performing hyper-parameter grid search (Géron, 2022) but also changing input variables such as sequence length or selected features should be conducted to achieve optimal performance of the model. As the dataset provided by the energy distribution industrial partner may chance in the future, the hyper-parameter optimisation was not considered a priority for this study given its high demand for high-performance computing resources. Another suggestion for future work is to consider a feature importance analysis, which allows for an optimal selection of features including only those which have a positive impact on performance. Another potential improvement could be achieved by introducing LSTM models with multiple parallel inputs and multi-step output (Brownlee, 2018). This approach would mean developing a single model for each group of similar households, using the energy consumption of multiple households as input, but also as output. In order to improve the precision of error assessment, the LSTM method deployed in this study should be evaluated in a scenario, where missing values are randomly distributed, rather than the step-wise prediction of values tested here. Also, the analysis of the results in this study is of descriptive nature. Statistical tests, which examine the distribution of the performance metrics of two performed models are another relevant contribution in future work for the INFINEED project, once the datasets are considered final.

8 Conclusions

In this study, an LSTM network is examined to predict electricity consumption. To be specific, it aims to address the problem of missing values both in historical and future data for a period of 1 day. First, a univariate LSTM model was developed based on historical energy consumption data. Then, different features such as the weekday, the outdoor temperature or the energy consumption of similar users were added to the model to create a multivariate LSTM. Finally, the two methods were trained and evaluated using a dataset of 212 households at daily intervals over three years. Both LSTM models performed better compared to the established benchmark calculated by averaging the consumption on the same weekday over the past two weeks. Among the LSTM models, the univariate model exhibited a marginal advantage over the multivariate model. When focusing on different household groups based on energy consumption volume as well as seasonality, both the univariate and multivariate LSTM models show strong predictive performance, especially for households with high energy consumption. This is particularly important because these households play a critical role in efforts to reduce electricity consumption. Findings from this study are directly relevant for the INFINEED project. Moreover, the analysis software framework along with the developed machine learning solutions represent not only a base solution required for addressing one of the INFINEED project goals but also represent a cornerstone for future work.

References

Afrifa-Yamoah, E., Mueller, U. A., Taylor, S., & Fisher, A. (2020). Missing data imputation of high-resolution temporal climate time series data. *Meteorological Applications*, *27*(1), e1873. https://doi.org/10.1002/met.1873

Axpo. (n.d.). *Electricity market facts and figures*. Retrieved June 1, 2023, from https://www.axpo.com/ch/en/about-us/media-and-politics/power-market-switzerland.html

Banachewicz, K., & Massaron, L. (2023). *The kaggle workbook: Self-learning exercises and valuable insights for kaggle data science competitions*. Packt Publishing Ltd.

Bourdeau, M., Zhai, Q. X., Nefzaoui, E., Guo, X., & Chatellier, P. (2019). Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustainable Cities and Society*, *48*, 101533. https://doi.org/10.1016/j.scs.2019.101533

Brownlee, J. (2017). *Long short-term memory networks with python: Develop sequence prediction models with deep learning*. Machine Learning Mastery.

Brownlee, J. (2018). *Deep learning for time series forecasting: Predict the future with mlps, cnns and lstms in python*. Machine Learning Mastery.

Chakraborty, D., & Elzarka, H. (2019). Advanced machine learning techniques for building performance simulation: A comparative analysis. *Journal of Building Performance Simulation*, *12*(2), 193–207. https://doi.org/10.1080/19401493.2018.1498538

Chollet, F. (2021). *Deep learning with python*. Manning Publications Co.

Cullen, D. (2011). Climate change. *Nature*, *479*, 267–268. https://doi.org/10.1038/479267a

Divina, F., Garcia Torres, M., Goméz Vela, F. A., & Vazquez Noguera, J. L. (2019). A comparative study of time series forecasting methods for short term electric energy consumption prediction in smart buildings. *Energies*, *12*(10), 1934. https://doi.org/10.3390/en12101934

Durand, D., Aguilar, J., & R-Moreno, M. D. (2022). An analysis of the energy consumption forecasting problem in smart buildings using lstm. *Sustainability*, *14*(20), 13358. https://doi.org/10.3390/su142013358

Ecoplan. (2015). *Smart metering roll out - kosten und nutzen*. Retrieved June 6, 2023, from https://www.ecoplan.ch/download/smmu_sb_de.pdf

Fu, Y., Li, Z., Zhang, H., & Xu, P. (2015). Using support vector machine to predict next day electricity load of public buildings with sub-metering devices. *Procedia Engineering*, *121*, 1016–1022. https://doi.org/10.1016/j.proeng.2015.09.097

Géron, A. (2022). *Hands-on machine learning with scikit-learn, keras, and tensorflow*. O'Reilly Media, Inc.

Guarnaccia, C., Mastorakis, N. E., Quartieri, J., Tepedino, C., & Kaminaris, S. D. (2017). Development of seasonal arima models for traffic noise forecasting. *MATEC Web of Conferences*, *125*, 05013. https://doi.org/10.1051/matecconf/201712505013

Holzer, A., Kocher, B., Bendahan, S., Vonèche Cardia, I., Mazuze, J., & Gillet, D. (2020). Gamifying knowledge sharing in humanitarian organisations: A design science journey. *European Journal of Information Systems*, *29*(2), 153–171. https://doi.org/10.1080/0960085X.2020.1718009

Holzer, A., Weber, S., Kocher, B., & Farsi, M. (2022). The interplay of feedback and incentive effects on electricity demand.

Horowitz, C. A. (2016). Paris agreement. *International Legal Materials*, *55*, 740–755. https://doi.org/10.1017/ilm.2016.46

Hwang, J., Suh, D., & Otto, M.-O. (2020). Forecasting electricity consumption in commercial buildings using a machine learning approach. *Energies*, *13*(22), 5885. https://doi.org/10.3390/en13225885

John, C., Ekpenyong, E. J., & Nworu, C. C. (2019). Imputation of missing values in economic and financial time series data using five principal component analysis approaches. *CBN Journal of Applied Statistics (JAS)*, *10*(1), 51–73. https://doi.org/10.33429/Cjas.10119.3/6

Kim, T., Ko, W., & Kim, J. (2019). Analysis and impact evaluation of missing data imputation in day-ahead pv generation forecasting. *Applied Sciences*, *9*(1), 204. https://doi.org/10.3390/app9010204

Lavin, A., & Klabjan, D. (2015). Clustering time-series energy data from smart meters. *Energy efficiency*, *8*, 681–689. https://doi.org/10.1007/s12053-014-9316-0

Lin, Z., Cheng, L., & Huang, G. (2020). Electricity consumption prediction based on lstm with attention mechanism. *IEEJ Transactions on Electrical and Electronic Engineering*, *15*(4), 556–562. https://doi.org/10.1002/tee.23088

Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (3rd ed.). John Wiley & Sons.

Martinez-Luengo, M., Shafiee, M., & Kolios, A. (2019). Data management for structural integrity assessment of offshore wind turbine support structures: Data cleansing and missing data imputation. *Ocean Engineering*, *173*, 867–883. https://doi.org/10.1016/j.oceaneng.2019.01.003

Meteostat. (2022). *Meteostat Python Library*. Meteostat. Retrieved June 6, 2023, from https://dev.meteostat.net/python/

Nielsen, M. A. (2015). *Neural networks and deep learning*. Determination Press.

Ozdemir, S. (2022). *Feature engineering bookcamp*. Manning.

Peppanen, J., Zhang, X., Grijalva, S., & Reno, M. J. (2016). Handling bad or missing smart meter data through advanced data imputation. *2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 1–5. https://doi.org/10.1109/ISGT.2016.7781213

Perret, L., Chevillat, Y., Wyrsch, N., Bloch, L., Holweger, J., Weber, S., & Péclat, M. (2018). Déterminer le potentiel de flexibilisation de la demande d'électricité des ménages. https://www.aramis.admin.ch/Default?DocumentID=50153&Load=true

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Proceedings*

of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 338–342. https://doi.org/10.48550/arXiv.1402.1128

scikit-learn. (2023). *scikit-learn/preprocessing/_data.py*. scikit-learn. Retrieved June 6, 2023, from https://github.com/scikit-learn/scikit-learn/blob/364c77e04/sklearn/preprocessing/_data.py#L270

Somu, N., MR, G. R., & Ramamritham, K. (2020). A hybrid model for building energy consumption forecasting using long short term memory networks. *Applied Energy*, *261*, 114131. https://doi.org/10.1016/j.apenergy.2019.114131

Wang, C., Baratchi, M., Bäck, T., Hoos, H. H., Limmer, S., & Olhofer, M. (2022). Towards time-series feature engineering in automated machine learning for multi-step-ahead forecasting. *Engineering Proceedings*, *18*(1), 17. https://doi.org/10.3390/engproc2022018017

Wang, M.-C., Tsai, C.-F., & Lin, W.-C. (2021). Towards missing electric power data imputation for energy management systems. *Expert Systems with Applications*, *174*, 114743. https://doi.org/10.1016/j.eswa.2021.114743

Weber, M., Turowski, M., Çakmak, H. K., Mikut, R., Kühnapfel, U., & Hagenmeyer, V. (2021). Data-driven copy-paste imputation for energy time series. *IEEE Transactions on Smart Grid*, *12*(6), 5409–5419. https://doi.org/10.48550/arXiv.2101.01423

Yang, J., Tan, K. K., Santamouris, M., & Lee, S. E. (2019). Building energy consumption raw data forecasting using data cleaning and deep recurrent neural networks. *Buildings*, *9*(9), 204. https://doi.org/10.3390/buildings9090204

Yilmaz, S., Weber, S., & Patel, M. (2019). Who is sensitive to dsm? understanding the determinants of the shape of electricity load curves and demand shifting: Socio-demographic characteristics, appliance use and attitudes. *Energy Policy*, *133*, 110909. https://doi.org/10.1016/j.enpol.2019.110909

Zhang, Y., Guo, L., Li, Q., & Li, J. (2018). Electricity consumption forecasting method based on mpso-bp neural network model. *Advances in Computer Science Research*, *50*, 674–478. https://doi.org/10.48550/arXiv.1810.08886

Zheng, A. (2015). *Evaluating machine learning models: A beginner's guide to key concepts and pitfalls*. O'Reilly Media. https://doi.org/10.1002/tee.23088

Zheng, T., Zhang, Y., & Fan, C. (2018). Research on hospital operation index prediction method based on pso-holt-winters model. *Proceedings of the 2nd International Conference on Computer Science and Application Engineering*, *23*, 1–8. https://doi.org/10.1145/3207677.3278092