Chair of Geoinformatics
TUM School of Engineering and Design
Technical University of Munich

Modeling Residential Energy Load Profiles with Semantic 3DCity Models and Machine Learning

Modellierung von Energieverbrauchsprofilen für Haushalte mit semantischen 3D-Stadtmodellen und Machine Learning

Scientific work for obtaining the academic degree

Bachelor of Science (B.Sc.)

at the TUM School of Engineering and Design of the Technical University of Munich

Supervisor Univ.-Prof. Dr. rer. nat. Thomas H. Kolbe
Chair of Geoinformatics

Advisor Khaoula Kanna, M.Sc.
Chair of Geoinformatics

Submitted by Julia Endo Barbosa
Tumblingerstr. 17
80337 München

Submitted on January 30, 2025 in München

Abstract

The residential sector accounts for a significant share of the final energy consumption in Germany. Therefore, the demand for accurate electricity consumption forecasting is a strong research topic. This thesis explores the integration of semantic 3D city models with machine learning to simulate residential electricity consumption at the household level. By leveraging CityGML, an international standard for 3D city modeling, and employing deep learning architectures, particularly Long Short-Term Memory (LSTM) networks, the study aims to improve the accuracy of predictions.

The structured methodology incorporates geospatial data, statistical data, historical electricity consumption records, and machine learning algorithms to identify patterns in residential energy use. A key innovation of this work is the combination of semantic city models with deep learning approaches, providing adaptability for multiple cases.

The thesis extended the use of the official load profile calculation method for households with missing information. Also, the results demonstrate that the LSTM model is promising and can improve prediction accuracy, offering valuable insights for the electricity industry. The findings support the potential of integrating geoinformatics and artificial intelligence for urban simulations.

Zusammenfassung

Der Wohnsektor macht einen erheblichen Anteil am Endenergieverbrauch in Deutschland aus. Deshalb ist eine genaue Vorhersage des Stromverbrauchs ein wichtiges Forschungsthema. Diese Arbeit untersucht die Integration semantischer 3D-Stadtmodelle mit Machine Learning, um den Stromverbrauch von Haushalten zu simulieren. Durch die Nutzung von CityGML, einem internationalen Standard für 3D-Stadtmodellierung, und den Einsatz von Deep-Learning-Architekturen, insbesondere Long Short-Term Memory (LSTM)-Netzwerken, soll die Genauigkeit der Vorhersagen verbessert werden.

Die strukturierte Methodik kombiniert geospatiale Daten, statistische Informationen, historische Stromverbrauchsdaten sowie Machine-Learning-Algorithmen, um Tendenzen im Energieverbrauch von Haushalten zu erkennen. Eine zentrale Innovation dieser Arbeit ist die Anpassungsfähigkeit der Kombination von semantischen Stadtmodellen mit Deep-Learning-Methoden.

Die Arbeit erweitert die Nutzung der offiziellen Lastprofilberechnungsmethode für Haushalte mit fehlenden Informationen. Die Ergebnisse zeigen, dass das LSTM-Modell vielversprechend ist und die Prognosegenauigkeit verbessern kann, wodurch wertvolle Erkenntnisse für die Energiebranche gewonnen werden. Die Ergebnisse unterstreichen das Potenzial der Integration von Geoinformatik und Künstlicher Intelligenz für urbane Simulationen.

Contents

Chapter 1

Introduction

1.1 Motivation

In 2022, 25,8% of the final energy consumption in Europe was caused by the residential sector, with its primary use being for heating purposes accounting for 63,5%, followed by electric devices and water heating. The categorization of energy use by fuel type shows a 30,9% coverage by natural gas, 25,1% by electricity, and 22,6% by renewables [1] . In Germany, the amount of energy generated by renewable sources increased by 6.7% from 2022 to 2023, resulting in 56% of the total energy production in 2023 [2]. In the next decades, there will be a transformation in the energy sector due to changes in consumption patterns, rising electro-mobility, electricity-powered devices, the increase of the earth's population to 10 billion humans, new social conditions like working from home, as well as self-generated electricity through photovoltaic panels [3].

In Germany, it is established by the "Energiewirtschaftsgesetzes" (Energy Industry Act) that the main goals of the Electricity and Gas Supply Systems are economic efficiency, environmental compatibility, and supply security [4] [5]. In 2023, in Germany, the installed power grid capacity was 165 GW, and the average load was about 52.5 GW, meaning that the capacity was 3 times larger than the load [6]. A high-quality energy consumption modeling can improve load planning, accommodate peak loads, and decrease the frequency of outages, resulting in better management and performance within this industry [7]. Moreover, energy consumption simulations support projects about sector coupling and integrating electric vehicles and heat pumps into the networks [8]. In addition, Bunn and Farmer estimated that a reduction of the forecasting error by 1% would save 10 million pounds in operating costs per year in the UK. This shows that sellers and buyers of electricity, grid, power plants, and storage operators would profit from accurate load forecasting [9]. For those reasons, it is clear that electricity consumption in the residential sector needs to be accurately modeled. However, electricity demand modeling and data are still outdated and incomplete [10]. Because of this, those models need to be adapted for the current and future scenarios and peak loads [11].

To forecast residential electricity and heat consumption, multiple countries use a single electricity consumption profile, known as standard load profile (SLP) [3]. In Germany, the household load profile H0 SLP developed by the German Federal Association of Energy and Water Management (Bundesverband der Energie- und Wasserwirtschaft e. V. (BDEW)) has been used for more than 20 years and is standardized and scaled based on annual consumption. The H0 SLP represents the energy consumption pattern of a household every 15 minutes, replacing a metered measurement. This load profile is scaled based on the household's

annual consumption and regardless of the number of inhabitants and apartment size because, according to the Load Profile Action Plan, the discrepancies were irrelevant [12][10]. In addition to that, the H0 SLP uses data measured in the 1970s or earlier [13]. Nevertheless, multiple intelligent meter data show that the actual energy and heat consumption deviate from the SLP by BDEW [14] [3]. The deviations supposedly came from the last decades' social, technical, and climatic developments [10] [14]. Yet, the standard load profiles are still used due to the lack of alternatives [10].

Researchers in this field rely on real data that energy companies do not make available due to privacy regulations and economic interests. To address the lack of data, 3D city models standards such as the CityGML Standard present an alternative solution, as they provide semantic information about city objects across entire urban areas. These models contain data about the covered volume, assignable area, building type, building usage, year of construction/building typologies, rehabilitation state, and number of inhabitants, all correlated to power, water, gas, and heat consumption. Given these correlations, simulations - such as the estimation of energy consumption, solar potential analysis, thermal remote sensing regarding heat emission of buildings, and utility network modeling- can benefit from the CityGML standard. Moreover, the standard allows simulations and computation models to be applied in different cities. For all these reasons, CityGML has become a central research focus [15].

Machine learning methods for modeling energy consumption are also receiving intensive research attention because of their capacity to analyze non-linear systems and extract knowledge from big datasets [16]. Widely implemented methods in this field are Artificial Neural Networks (ANN) and their variants, such as Feed Forward Neural Networks (FFNN), Recurrent Neural Networks (RNN), and Probabilistic Neural Networks (PNN) [17]. These Data-driven models deliver high forecasting accuracy and have been used to model energy consumption for short to long-term forecasting at residential, building, and country levels [9].

This thesis aims to contribute to a better management of Germany's energy system by calculating the electricity consumption of residential buildings. Then, the possibility of improving the load profile calculation through machine learning methods should be explored and compared to an existing Load Profile calculation method. Finally, the generated electricity consumption curve and its related information should be integrated into the CityGML.

1.2 Research Questions

To achieve this thesis's goal, research was done regarding Load Profiles in Germany, the use of CityGML models in electricity consumption calculation, and machine learning techniques for electricity consumption. This work addresses the following research questions:

1. Which electricity consumption Load Profile models exist, and how do they work?

2. How can CityGML Standard models be used for electricity consumption calculations?

3. How can Machine Learning techniques improve electricity consumption modeling?

1.3 Structure

The theoretical background will provide a brief overview of key concepts related to load profile calculation, CityGML Standard, and Machine Learning to better understand the choices that define the methodology. It is important to note that a deeper exploration of each topic is beyond the scope of this work. Following this, the literature review summarizes some related and relevant studies that provide context and support for this thesis.

The methodology section explains the step-by-step implementation. This means, how to use a CityGML file as input to first calculate for residential buildings the amount of households and their residents. From there, the yearly electricity consumption for every household in all residential buildings from the CityGML file is calculated through a specialized type of ANN, the Long-short term memory Network. The output of the ANN and the number of residents are then used as input for the chosen Load Profile Calculation method for comparison purposes.

Then, the visualization chapter explains how the results in semantic tables and consumption curves were integrated into the 3d Citydb web map client tool from the Technical University of Munich.

Finally, Chapter 6 discusses the results regarding the performance of the LSTM in predicting yearly energy consumption values for each household. Then, the conclusion and suggestions for further research finish the thesis.

Chapter 2

Theoretical Background

2.1 Residential Load Profile Model

A residential Load Profile is the representation of the electricity consumption of households over time. The models that calculate the electricity usage pattern are mainly categorized by having either a bottom-up or a top-down method. Still, they can also be differentiated by their sampling rate, application, or statistical techniques [3].

Bottom-up calculation methods aggregate the electricity consumption of each household appliance and the household occupant's usage pattern into the total household electricity consumption profile. For this kind of model, the total consumption can also be calculated through the characteristics of the house (e.g., size, building materials, heating/cooling characteristics) and weather conditions. Therefore, bottom-up models need highly detailed household or device-level data and deliver specific and adjustable results. They can be used to analyze the impacts of different technologies, policy decisions, or energy optimization techniques. It is also possible to aggregate the results to create a Standard Load Profile (SLP), a profile that can be used at a country level. The main benefit of this method is that historical data is not necessary, but on the other hand, bottom-up methods are computationally heavy [3].

Top-down models work the other way around, as they derive a single household profile from macro-level data. This data can be the historical energy consumption from a region, a country, and/or a sector. Then, assumptions are made between the macro-variables, such as socioeconomic indicators and weather conditions, and the total energy consumption data. Contrary to Bottom-up models, they require historical data but do not require highly detailed data about individual usage of electric appliances. They are not as computationally heavy but result in loss of information. In addition, there are also Hybrid models, which combine techniques from both [3].

Another way to categorize the load profiles is by their sampling rate, which can be low, middle, or high resolution. Low resolution refers to models with data collected at intervals longer than fifteen minutes. Middle-resolution models should have a sampling rate between fifteen and one minute. High-resolution models use information collected in intervals from one minute or shorter. Each sampling rate is adequate for different uses of the load profiles. For example, low-resolution models could study the impact of energy prices, and high-resolution models can accurately predict fluctuations in residential energy demand [3].

The residential load profile models can also be grouped by their application or statistical method. The main goals of calculating electricity consumption are demand-side manage-

ment, planning, controlling, and designing energy systems, distribution grids, and local energy efficiency strategies. The principal statistical methods are Markov chain, probabilistic, and Monte Carlo models [3]. More information about this can easily be found online and should not be mentioned within this thesis.

2.2 City Geography Markup Language (CityGML)

The City Geography Markup Language is an Extensible Markup Language (XML)-based international standard for representing, storing, and exchanging semantic 3D city models approved by the Open Geospatial Consortium (OGC). Semantic 3D city models differ from other virtual city models, which purely contain graphical and geometrical properties by also containing thematic classes, attributes, and their interrelationships. This enables multiple applications for CityGML, such as thematic queries, analysis tasks, spatial data mining, environmental and training simulations, urban planning, disaster management, vehicle and pedestrian navigation, and more. It is intended to develop a common definition and understanding of the entities, attributes, and relations within a 3D city model so all users in the community use the same representation standard, resulting in easy data exchange between different fields [18] [19] [20].

The CityGML standard is composed of its Conceptual Model and its Encoding Standard. The Conceptual Model defines how 3D urban objects should be modeled through classes and relationships, such as hierarchies or inheritance, concerning their geometrical, topological, semantical, and appearance properties. This model is modularized, meaning it has a Core module that is decomposed into other modules, such as buildings, roads, railways, tunnels, bridges, city furniture, water bodies, vegetation, and terrain. The conceptual model is expressed through diagrams using the Unified Modeling Language (UML) for a structure representation, where each module has its own schema. CityGML models can also be of different levels of detail (LoD), each being more coherent to other applications (see figure 2.1) [19].

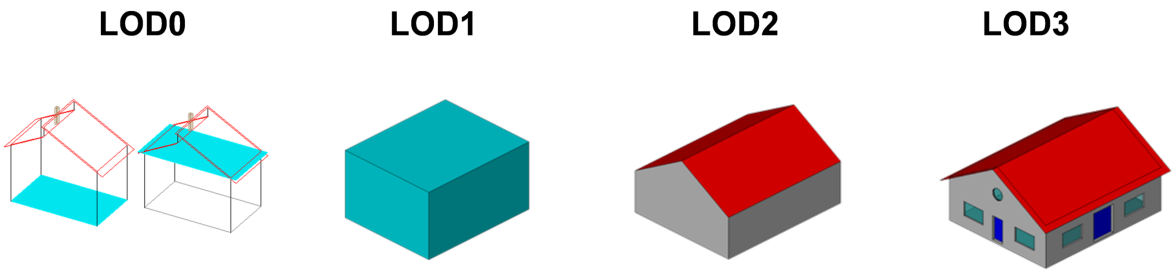


Figure 2.1: Representation of the same real-world building in the Levels of Detail 0-3 [19]

The spatial properties of CityGML objects are represented through geometries using ISO 19107. Geometry can have 0-, 1-, 2-, or 3-dimensions and can be formed by primitive geometries like single points, curves, surfaces, and solids, as well as by spatial aggregates and composites. 3D surfaces can be represented by 3D polygons or 3D meshes such as triangulated irregular networks (TINS) [19].

As mentioned before, the CityGML is also composed of its encoding standard that represents the conceptual model through XML notation, which follows the UML-to-GML applica-

tion schema encoding rules in Annex E of [21]. The conceptual model defines three types of objects: features, top-level features, and geometries. The features must have a mandatory featureID and can have an optional identifier. Within a CityGML dataset, the featureIDs are unique, and the identifiers have an identical value for all versions of the same real-world object, allowing them to be distinguished and referenced. In CityGML, it is essential that geometries also have a unique featureID. This enables, for example, referencing all geometries by their IDs so they can have a specific texture [21]. An example of an LoD 3 house in the CityGML version 2.0 can be found in the CityGML Wiki[22] to clarify the notation.

Furthermore, the CityGML Conceptual Model can be extended through Application Domain Extensions (ADEs) for specific applications or domains. They add new classes, attributes, and relations, allowing CityGML to meet the user's information needs while conserving its structure [19]. Some ADEs were applied, for example, for adding information on the gas consumption of a building, adding a new module for monuments, and for simulating noise immission. There is also the CityGML Energy ADE for adding features necessary to perform energy simulations and to store the results [23].

2.2.1 Representation of buildings with CityGML

It can be understood from the previous chapter that buildings in CityGML have thematic and spatial characteristics. They are represented by the top-level feature type building, and they can be logically subdivided into building parts, starting in the CityGML version 3.0, into storeys and building units (see figure 2.2). The geometry of a building is composed of surfaces, such as wall surfaces, ground surfaces, roof surfaces, and ceiling surfaces. Depending on the LoD, the buildings have roofs, doors, windows, balconies, chimneys, dormers, and more [19].

The encoding of the building concept through XML notation can be seen in the Appendix, with an example of a LoD2 house modeled by the CityGML 3.0 Standard. It is important to note that this Thesis was done using CityGML version 2.0, which does not have the subdivision "building units". The solution was to model the simulated households as "building parts".

2.3 Deep Learning

2.3.1 Introduction

Machine learning is a key part of modern technology, driving many everyday applications. It powers web searches, filters content on social media, provides personalized recommendations on shopping platforms, and enhances devices like cameras and smartphones. Deep learning, a subset of machine learning, has become increasingly popular due to its capability to handle complex and high-dimensional raw data through artificial neural networks more effectively than earlier methods [24].

Traditional machine learning was limited and relied on feature engineering, which manually makes data usable. Deep learning solves this, by automatically identifying useful patterns and insights directly from raw data. Using multiple layers, deep learning understands

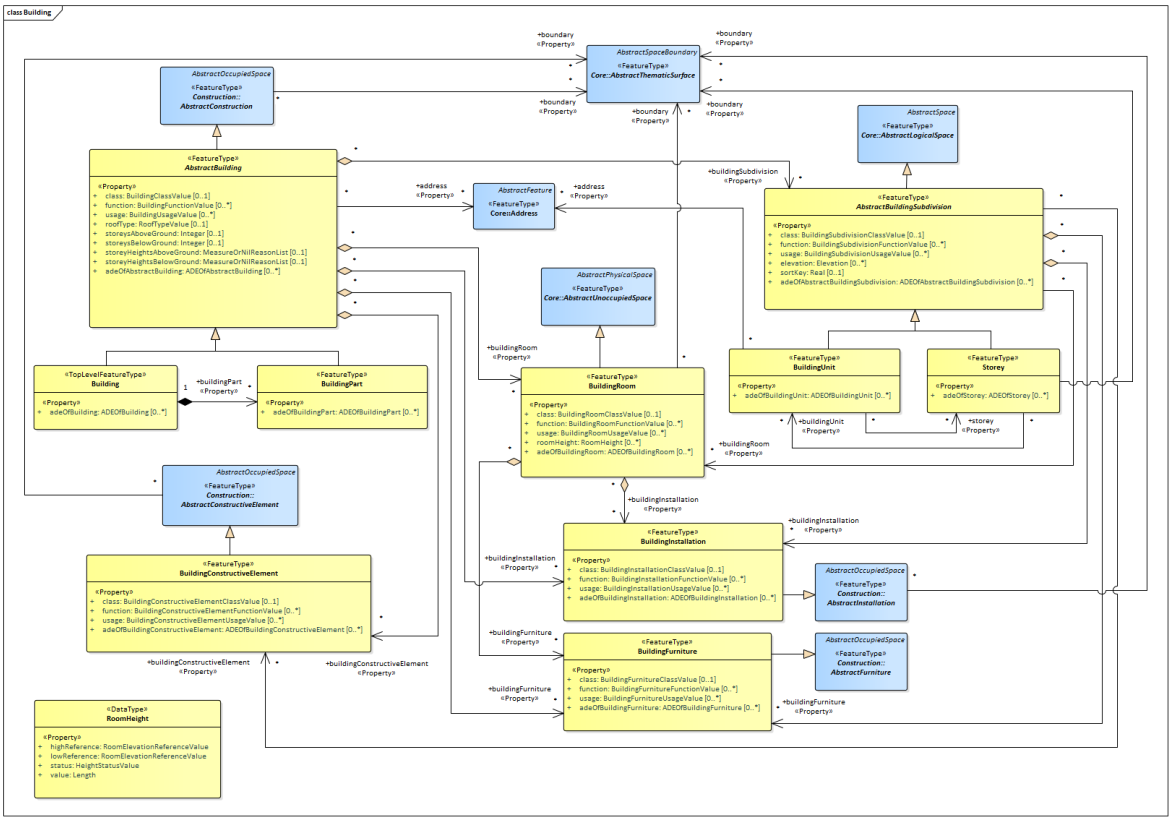


Figure 2.2: UML diagram of CityGML's building model [19]

complex relationships and solves difficult tasks. Furthermore, with powerful computational resources, such as GPUs, and the availability of large datasets, deep learning has achieved state-of-the-art performance across numerous domains [24].

This chapter delves into the basic theoretical background of artificial neural networks ANNS and explores a special type called Long Short Term Memory LSTM, later used in the thesis for predicting residential electricity consumption.

2.3.2 Artificial Neural Networks (ANNs)

"Artificial neural networks are computational models inspired by the nervous system of living beings." [16]. As mentioned before, these models can solve problems in many fields, for example, analysis of images, speech and writing pattern classification, face recognition with computer vision, control of high-speed trains, stock forecasting on the financial market, anomaly identification on medical images, and in the case of this thesis forecasting residential energy consumption. These problems are solved through different applications of the ANN, such as universal curve fitting (function approximation), pattern recognition/classification, data clustering, prediction systems, and more [16].

Inspired by the human brain, ANNs can receive, keep knowledge and learn from it. They are formed by layers of interconnected artificial neurons j (the processing units), which receive input signals, represented in figure 2.3 by x_1, x_2, x_3,..., x_n, from the external environment or previous layers. Each neuron has a relevance, represented by its weight w_{ij}. Then,

the activation function a_j, a mathematical function, defines the output and introduces non-linearity into the model. The activation functions can be the sigmoid, the tanh, the Rectified Linear Unit (ReLu), the SoftMax, and the E functions [25][16].

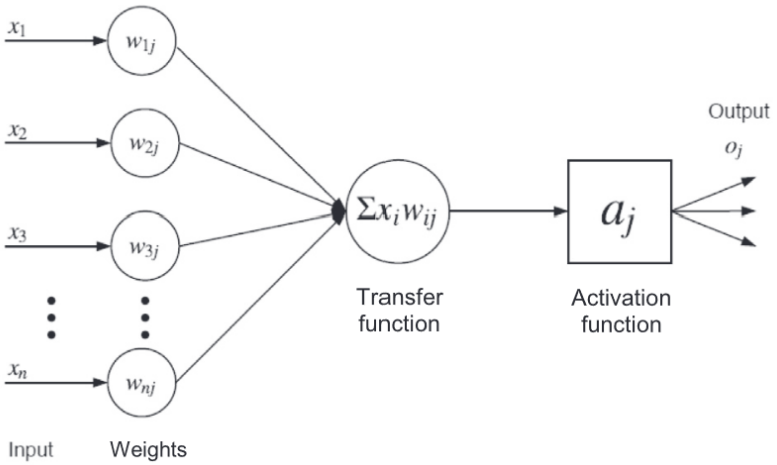


Figure 2.3: A single neuron with input x_i, weight w_{ij}, activation function a_j and output function o_j [9]

Non-linearity differentiates ANN from Linear Regression algorithms, enabling them to work with data with more complex relationships between variables. However, Linear Regression is interpretable and gives clear insights about the relations between variables, while with ANNs, the logic behind the model's decision cannot be explained, being then considered "black boxes.". In addition, it is a challenge to find the weights w_{ij} that deliver the best performance of the network while avoiding overfitting, learning not only the patterns but also the noise and underfitting, being too simple for the problem (see figure 2.4) [26].

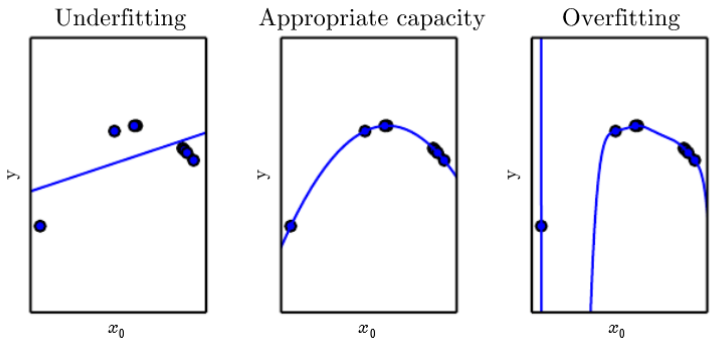
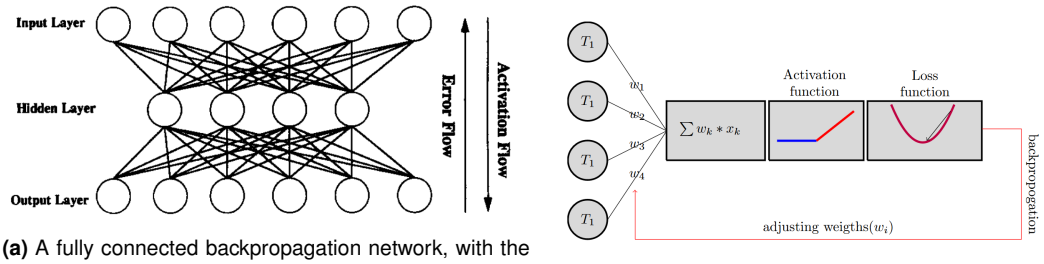


Figure 2.4: Representation of a Underfit, Overfit and Good Fit [27]

To fully comprehend ANNs, it is essential to understand the training and testing processes. During training, the model learns from the input data phase by generalizing solutions and generating quality output, and the testing process controls the quality of the network's performance. The data must be separated into training and testing sets to realize these. Assuming the available data comprises 1000 samples, the network can be trained on, for example, 700 samples and tested on the 300 remaining unbiased samples. These processes can be further explained using the simplest form of ANN, the Multilayer Perceptron (MLP) network [16].

An MLP can be classified as having a multiple-layer feedforward architecture, meaning they are formed by an input layer, at least one hidden layer, and an output layer. In this case, the information flows unidirectionally through the model to generate results. To create the model, the MLP is first trained through the Backpropagation algorithm [16].

The backpropagation algorithm (see figure 2.5a) has two types of phases, the Activation Flow and the Error Flow that are applied iteratively to the model. During the first one, the input data passes through the neurons and generates an output. Then, the error between the generated and expected output in the training dataset is calculated through a loss function (see figure 2.5b). The error is backpropagated to seek the global minima through gradient descent. During the Backpropagation phase, the neurons' weights are adjusted, improving the network's predictive accuracy (see figure 2.5b) [28][29]. After training, the previously separated testing dataset tests the model's performance.



(a) A fully connected backpropagation network, with the direction of activation and error flow indicated [28]

(b) Schematic of Artificial Neural Netowrks [29]

Figure 2.5: Schema of Neural Networks indication backpropagation

2.3.3 Network's Hyperparameters

Understanding the hyperparameters is necessary since they strongly impact the network's performance. In contrast to the parameters that can be learned and updated in the training process, the hyperparameters must be set before training the model[30]. Choosing optimal values for them is difficult, so they are often chosen based on literature recommendations, experience, trial and error, or alternatively through tuning strategies [31].

Some examples of hyperparameters are the batch size, number of epochs, and learning rate. Batch size determines how many samples the network should work through before updating the model's internal parameters. The number of epochs defines how many times the model should completely pass through the training data. The learning rate determines how often the weights are updated during training, defining how fast the model adapts to the problem (see figure2.6)[32].

2.3.4 Long-Short Term Memory Network

This subchapter introduces the theory behind LSTM Networks due to their outstanding results in recent research for forecasting residential energy consumption [17] [34] [35].

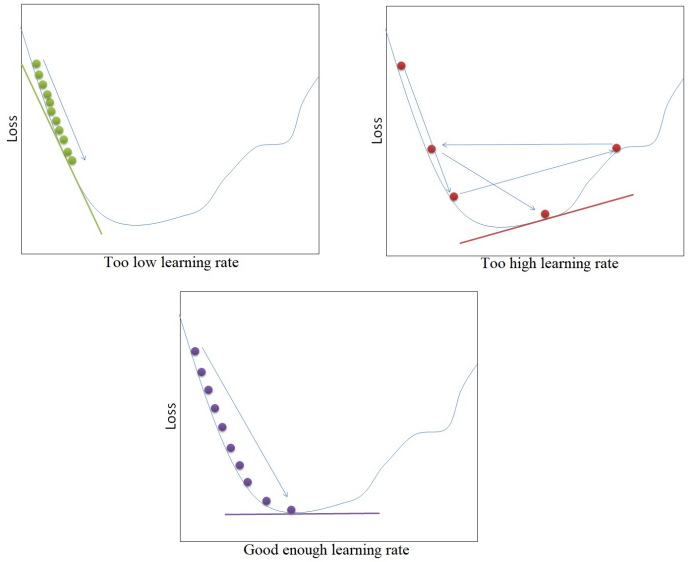


Figure 2.6: Learning rate ilustration [33]

The Long-Short-Term Memory (LSTM) Networks are a special type of ANNS. They belong to the Recurrent Neural Networks (RNN), which are designed for solving sequence problems. The difference between an MLP and an RNN is the addition of loops to the network's structure, while in MLPs, the information flows exclusively forward. For instance, neurons in one layer can pass signals laterally to each other or feedback their output to be treated as a new input vector (see figure 2.7). The loops provide the network with a memory, which helps to learn the ordered nature of input sequences[36].

RNNs can suffer from the vanishing gradient problem when the backpropagated gradient can tend to zero, interfering with the learning of the model. The structure of LSTMs partially solved this by introducing cells that contain a memory internal state, an inner loop, a forget, an update, and an output gate (see figure 2.7). Nevertheless, the exploding gradient problem where error gradients accumulate and result in very large gradients can still affect this type of ANN preventing it to learn and outputting valid values[36] [37].

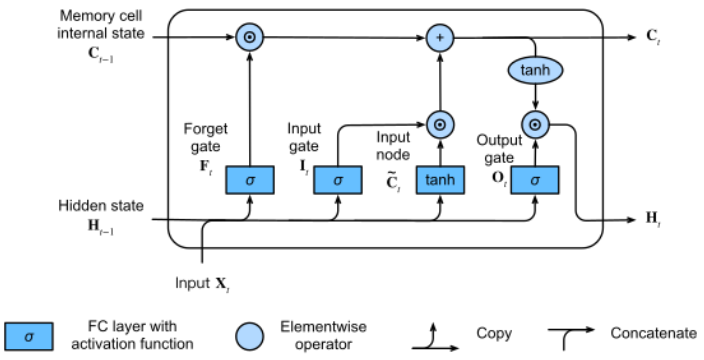


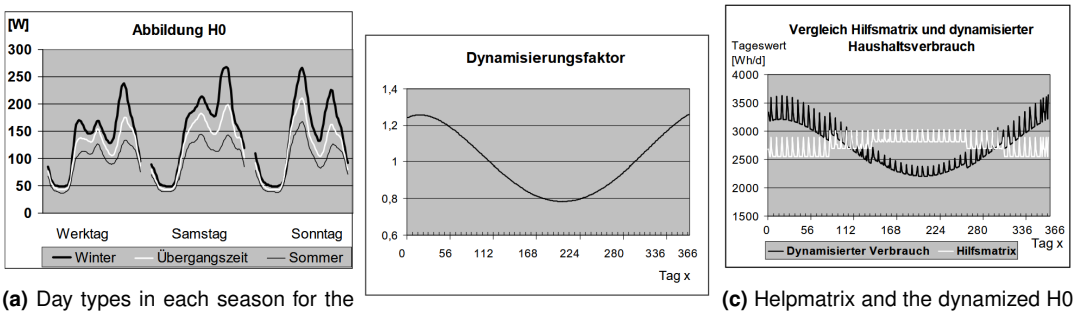
Figure 2.7: input gate, the forget gate, and the output gate in an LSTM model [38]

Chapter 3

Literature Review

3.1 Load Profile Models

This subchapter presents some of the residential Load Profiles developed for Germany. The official one from the German Federal Association of Energy and Water Management (Bundesverband der Energie- und Wasserwirtschaft e. V. (BDEW)) is the H0 Standard Load Profile (H0 SLP), created with the intention of helping energy companies to keep track of the residential energy consumption since this is normally not measured. The standardized profile was developed based on daily consumption curves with 15-minute timesteps for Winter, Summer, and Transition periods and for weekdays, Saturdays, and Sundays. Holidays have the same consumption profile as Sundays. The daily profiles are assigned to all days of the year, forming a "Help Matrix" (Vergleichsmatrix). This matrix is then multiplied by a dynamization factor, resulting in a more harmonized yearly profile (see figure 3.1). The finalized yearly profile is normalized to a total energy consumption of 1000 kWh. The basis for adjusting the SLP for a customer is his previous yearly energy consumption or a realistic forecast in MWh/a [39].



(a) Day types in each season for the H0 SLP

(b) Dynamization factor

(c) Helpmatrix and the dynamized H0 SLP

Figure 3.1: Elements that build the H0 SLP [39]

As already mentioned in the introduction, the H0 SLP has some conflicting characteristics. The data used for creating it were measured in the 1970s, and ninety percent consists of hourly measurements, while ten percent from fifteen minutes resolution [13]. In Addition, the H0 SLP did not consider the number of inhabitants or the household size because, according to its Action Plan, the differences were irrelevant [10] [12]. Nevertheless, these characteristics correlate to residential energy consumption [15]. Smart meter measurements show that this SLP fails to capture consumption trends. These deviations probably stem from social, technical, and climatic developments [10] [14]. Despite this, most energy suppliers still use the H0 SLP due to the lack of alternatives. [13] [10].

The DemandRegio Project, by the German Federal Ministry for Economic Affairs and Energy (Bundesministerium für Wirtschaft und Energie), aimed to address the insufficiency in Germany's energy and gas demand models [10]. This project calculates the final energy and gas demand for the residential, industrial, commerce, trade, and services subsectors in high temporal and spatial resolution. Four methods are proposed to calculate regional energy consumption from country-level values: a Top-Down and a Bottom-Up, each with spatial and temporal resolution [40].

Since the spatial resolution methods calculate a regional annual consumption value and the temporal resolution method calculates an energy consumption time series, which is more related to the scope of this thesis, only the method presented in 3.4.2 in [40] is relevant to this literature review.

The Zeitverwendungserhebung (Time Use Survey) method is a Bottom-Up approach based on activity-based electricity consumption. The EnergieAgentur.NRW GmbH provides information about the share of electricity used for each appliance by household size (see figure 3.2). The time-use survey provides information about the presence at home, the availability of daylight based on geographic latitude and longitude, and assumptions about sleeping hours. To derive the ZVE load profile, the total regional energy consumption should first be divided by the household size and then multiplied by the activity and base load-related intensities. The result for a winter day in Berlin, München, and Jülich can be seen in Figure ref3.3 [40].

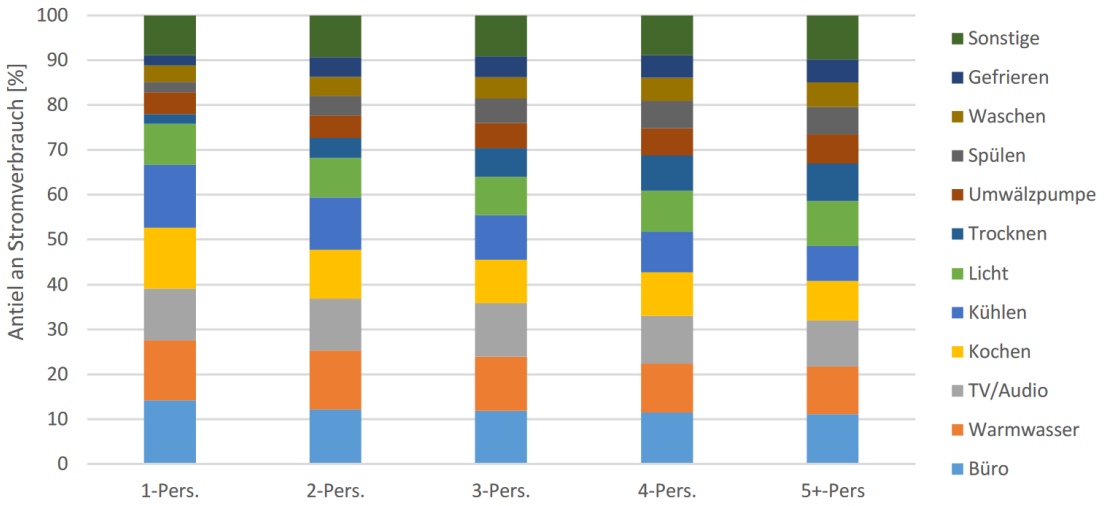


Figure 3.2: Activity intensities of private households based on data from the Time Use Survey 2012 [40]

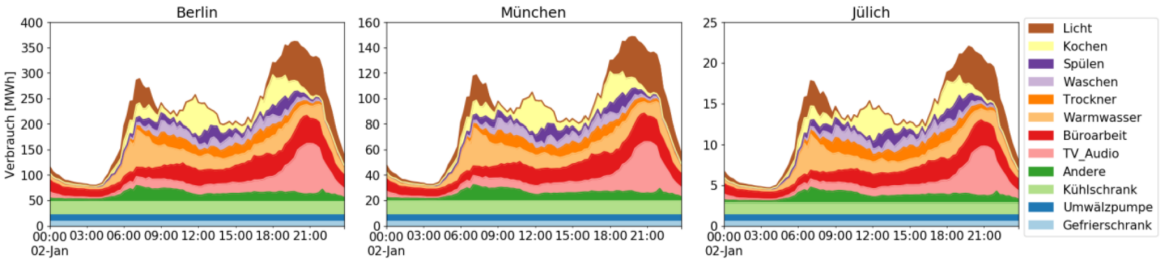


Figure 3.3: ZVE load profiles for a winter day in three locations: Berlin, Munich, and Jülich [40]

The study "Data-Driven Load Profiles and the Dynamics of Residential Electric Power Consumption" by Anvari et al. focused on creating an accurate, high-resolution, easy-to-use load profile to be developed by understanding the dynamics of residential electricity consumption [13]. The researchers analyzed high-temporal resolution power consumption data from Austrian and German households and noticed extreme consumption spikes, which are not considered by the H0 SLP. The proposed method consists of an easily applicable and adaptive methodology for creating residential load profiles. First, the Averaged Load Profile (ALP) is extracted through empirical mode decomposition, separating the long-term consumption trends from short-term fluctuations. Through superstatistics, a stochastic model is developed to describe the fluctuations, which, within a time window of circa 1000 seconds, follow a distinct Maxwell-Boltzmann distribution. Finally, the combined techniques result in a data-driven load profile (DLP) that fits the real consumption curves better than the H0 SLP.

3.2 Energy consumption calculation with 3D City Models

3D City Models are suitable for energy-related calculations as they include semantic and geometric information about city objects such as building usage, year of construction, volume, area, building height, and more [15]. Therefore, this subchapter presents some work that used the CityGML Standard for energy-related purposes.

Robert Kaden's dissertation [41] focuses on using semantic 3D city models to support strategic environmental and energy planning in urban areas. He introduces the "Energy Atlas," a platform that integrates 3D City Models and geospatial and statistical data, which also simulates building-specific and large-scale heat, electricity, and hot water demand. The calculations are based on statistically derived averages. The input values and the calculated energy demands are modeled with CityGML and a specified Energy Application Domain Extension (EnergyADE).

Chapter 4 explains how the input parameters for the calculations were determined, using Berlin as the primary case study but also exploring the applicability to other regions. The dissertation presents methods to calculate energy-relevant values at the building level. Many parameters were addressed, for example, heat-transmitting surfaces, volume, area, age classes, thermal properties, potential renovations, number of households, and number of residents [41].

The most relevant parameters for this thesis are the number of households and residents. For calculating the number of units in a building from a specific age class, the method assumes this is correlated to the building's volume and combines it with the information of a site survey. This survey of a representative selection for Berlin of 374 buildings collected information about the real number of residential units and other energy-relevant parameters. The method for calculating the number of residents per unit also assumed a strong correlation to the building's volume. Due to privacy reasons, building-level data about the number of residents is not available but rather at the statistical block scale. That is why the population of a statistical block is first divided to the buildings and then distributed among the estimated building units. For realizing the second step, statistical data from the office for statistics Berlin-Brandenburg provides the share of household sizes (one to three and multi-person) in each city district. After determining the number of units, residents, and proportion of household sizes, the third step is to assign the residents to the households

to achieve the proportions derived from the statistical data. The method-estimated average number of residents per residential unit was 1.58, differing only 0.07 from the official value from the Berlin-Brandenburg Statistics Office [41].

$$R_i = \frac{R_k \cdot V_i}{\sum_{i=1}^{n} V_i} \tag{3.1}$$

where:

- R_k is the number of residents in a statistical block k;
- V_i is the building's volume [m³];

The following chapter details the methodology for calculating heat, electricity, and hot water energy demand. Due to the scope of this thesis, only the electricity-related section 5.1.2 is relevant. For estimating the building's energy demand caused by electricity, the previously determined number of residents per household is incorporated with statistical data on appliance usage [41].

$$W_i = \sum_{k=1}^{4} n_k \cdot W_k \tag{3.2}$$

where:

- W_i - the electricity consumption in [kWh];
- n_k - the number of households with k people;
- W_k - the average electricity consumption per household with k people.

Köhler et al. conducted a similar study to estimate the number of households and their residents through CityGML [8]. The building volume and heated area are derived from the 3D model. Next, the number of households is calculated through statistical data. Based on statistical data on frequency density distribution, the total heated area of the building is distributed into the the households, and then the number of occupants per household is determined based on the previously distributed household area. The methodology was validated across three German counties representing urban, suburban, and rural areas by summing the results at the county level and comparing them with official records of the total population and household counts. The results were satisfactory, since on the county level, the results for the number of households diverge by less than 7% and for the total population by less than -14%. It was highlighted that the divergence of the result could be partly due to the usage of statistical data from 2011, which means the household distributions and population growth might be unaccounted for. Also, the statistical data used was at the country level, which normally results in an overestimation of the population for rural areas and an underestimation of dense cities.

An older study, also by Köhler et al. for generating load profiles is used in the simulation platform SimStadt from HFT Stuttgart. The study aims to generate synthetic load profiles with 3D CityGML as input data, taking stochastics and randomness into account. The method applies to building and city quarter levels. This work is not open source and therefore, further details are out of the scope of this thesis [42].

3.3 Energy Consumption calculation with ANN

There are many studies on the prediction of electric power consumption with many different ANN architectures available. Due to the recent success of LSTMs in dealing with time series data and the scope of this thesis, only studies that mainly studied this type of ANN are reviewed.

The paper “Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Networks” aims to address the main challenge in this task, which is the lack of consistent consumption patterns for individual households [35]. The dataset for training and testing the LSTM was data measured with smart meters for thousands of individual households provided by the Smart Grid Smart City project. Other methods, such as conventional back-propagation neural networks and k-nearest neighbor regression, are tested with different hyper-parameters and time horizons for 69 households. When compared, the LSTM showed the best overall forecasting performance.

Hyeon et al. compare three types of ANNs, a vanilla LSTM, a sequence-to-sequence, and a sequence-to-sequence with attention mechanism, to evaluate the latest deep learning models. Using the dataset from the UCI machine learning repository from a household in France, the active energy was calculated and then averaged over an hour. Three years of data were used for training and one year for testing. When comparing the different models in nine different time horizons, the LSTM performed best in all cases [43].

The study "Predicting Household Electric Power Consumption Using Multi-step Time Series with Convolutional LSTM" uses a hybrid deep learning model combining Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) networks to forecast electricity consumption for a single household. The data used was also from the UCI machine learning repository. The implementation consists of two phases; first, the upcoming 500 hours of global active power is predicted through an LSTM, while in the second phase, a ConvLSTM predicts weekly electricity consumption through a time series provided by smart meter measurements and captures spatial characteristics. The evaluation, also through RMSE, shows outperforming results compared to the other experimented models that can be implemented for, e.g., smart grid planning [44].

Chapter 4

Methodology

4.1 Overview

This work aims to find alternatives to simulate residential electricity consumption in Germany in order to contribute to the management of its energy system. Semantic 3D city models from the CityGML Standard, statistical data on household sizes in Germany and average electricity consumption based on household size, open smart meter data from Munich from [45], and an LSTM network were used to model the electricity consumption on the household level in residential buildings in the neighborhood "Harthof" in Munich.

The CityGML dataset provided information on the area, the geographical location, and the building geometry. This was used to estimate the number of residents and area at the household level. Then, the time series data measured with smart meters for households in Munich with different addresses and areas, available in [45], was used to create an LSTM neural network that predicts the electricity consumption at the household level. For comparison reasons, two consumption curves were generated with the H0SLP method from [39]. The results are visualized in a 3D Viewer. The overview of this methodology can be seen in Schema 4.2.

The City of Munich provided the dataset used in this thesis, which consists of the neighborhood "Harthof," containing 383 buildings, of which 277 are residential (see Figure 4.1). The methodology was also tested using an open-source data set from geodaten.bayern.de. In this case, a 2km x 2km tile in the city district Isarvorstadt was downloaded. Refer to the GitHub repository [46] README file for a step-by-step guide on how to implement this methodology.

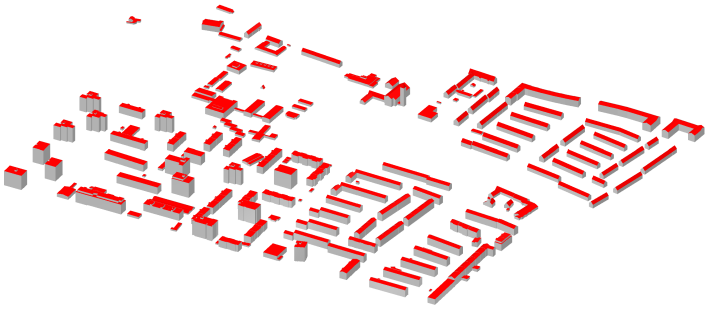


Figure 4.1: Harthof dataset visualised in the software Kit Model Viewer

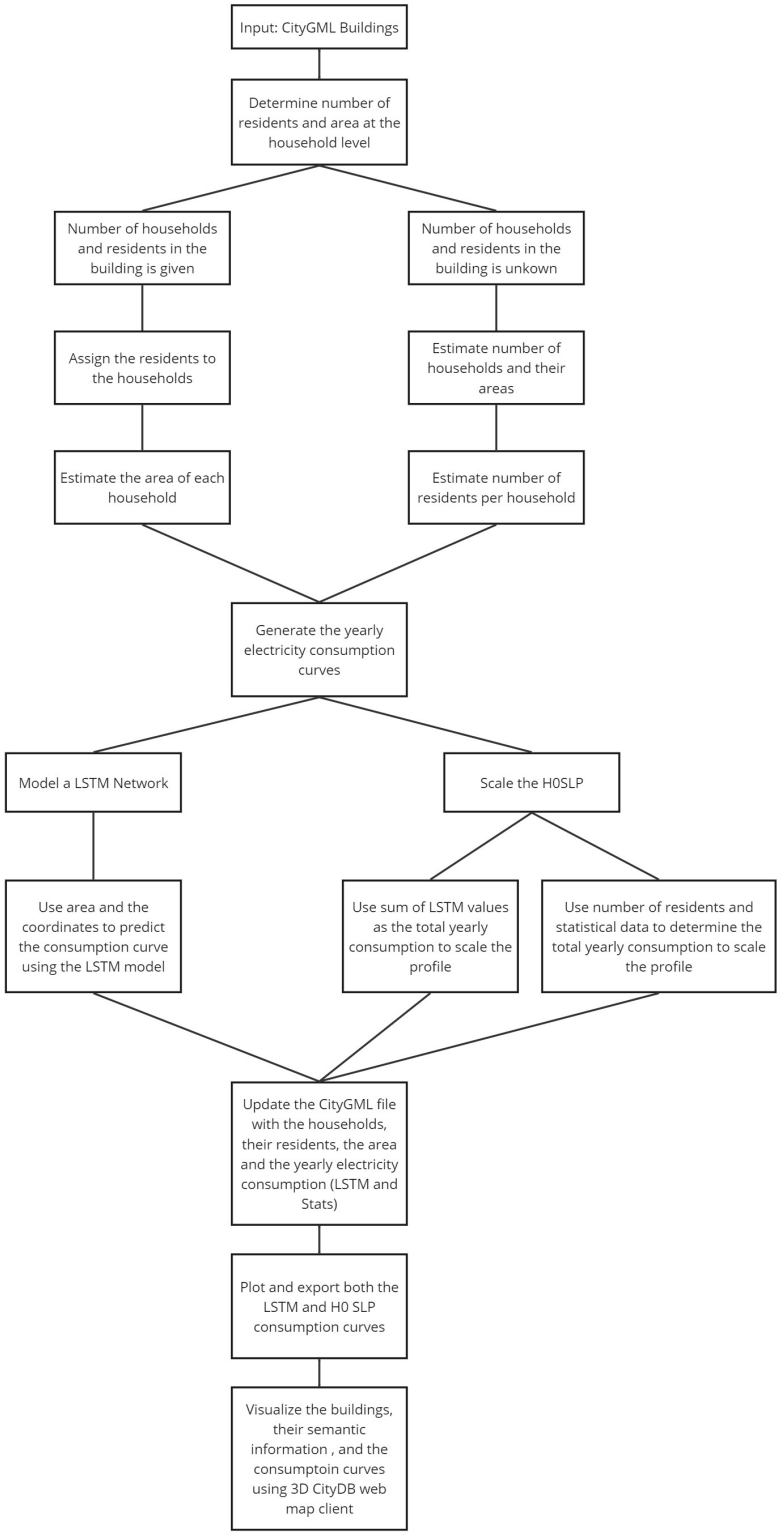


Figure 4.2: Schema resuming the methodology steps

4.2 Estimating households and their residents

4.2.1 Case 1: Given number of households and residents in the building

Even though CityGML files follow a specific structure, specific information about buildings in the CityGML dataset may not always be available. It is more common that information about the number of households and residents in buildings is not provided due to privacy reasons, like in the data from the Bavarian Mapping Agency. Nevertheless, for this thesis, the Harthof dataset provided this information for some buildings. Assuming that the number of households and residents in the input file is correct and up-to-date, it was a priority to respect the given real data by only distributing the given number of residents to the households and not to estimate these numbers from scratch. To do this, chapter 4.3.5 of Robert Kaden's [41] dissertation was used. The first part of the chapter in [41] defines the total number of residents in the building, information which, for this case, is already given, making only the second step relevant.

In this thesis, the buildings are assigned to Munich's city districts through the easting and northing coordinates provided by the CityGML dataset. Then, the residents are distributed to the households based on statistical data from the city of Munich [47] about the share of household size in each city district. For the city district Harthof, the ratios can be observed in Table 4.1.

City district/Number of residents	1	2	3	4	5 or more
Harthof	53,4%	24,24%	10,6%	7,8%	4,0%

Table 4.1: Share of private households in percent as of December 31, 2020, by household size in the city districts [47]

First, the number of one, two, three, four, and five-person households represented by NH_i is determined by multiplying the total number of households with the statistical ratios (see equation 4.1). This leads to a number of residents $NR_{statistics}$ calculated based on the statistical information (see equation 4.2).

$$NH_i = H \cdot r_i, \quad i = 1, \ldots, 5 \tag{4.1}$$

where:

- NH_i - the number of households with size i;
- H - total number of households in the building;
- r_i - statistical ratio for each household size in a specific city district.
- i - household sizes.

$$NR_{statistics} = \sum_{i=1}^{5} NH_i \cdot i, \quad i = 1, \ldots, 5 \tag{4.2}$$

where:

- $NH_{statistics}$ - number of residents in the building based on the statistics;
- NH_i - the number of households with size i;
- i - household sizes.

Next, the real number of residents NR_{input} is distributed among the households NH_i while maintaining the ratio of the statistical data and from the real data from the input CityGML as accurately as possible. To do this, the difference between NR_{input} and $NR_{statistics}$ is calculated and used as a reference for using table 4.2, which redistributes the number of households with certain sizes. For example, if the difference between NR_{input} and $NR_{statistics}$ is 2 people, the building should have one less household with one resident and one more household with 3 residents.

Difference [People]	1 Person Household	2 People Household	3 People Household	4 People Household
1	-1	+1		
2	-1		+1	
3	-1			+1
4	-2	+1		+1
5	-2		+1	+1
6	-3	+1	+1	+1
7	-4	+2	+1	+1
8	-5	+3	+1	+1

Table 4.2: Scheme for determining the number of households of different sizes by distributing the difference in persons [41]

Differences in the available statistics from Munich and Berlin result that the table 4.2 only edits for households from 1 to 4 people, while in Munich, the statistical data [47] provided information about households from 1 to 5+ people. This means a more suitable table covering the fifth household size is needed. Nevertheless, this thesis used the dissertation's table to redistribute the households because households with 5 or more people represent only 3,1% of Munich's total households [47], resulting in many buildings with no households with this size and, therefore, no need for redistributing them.

Another problem needed to be addressed due to adapting an existing method to this thesis case study. The table 4.2 only addresses a difference between NR_{input} and $NR_{statistics}$ from 1 to 8 people, but in the Harthof case, some cityGML buildings came with an incoherent number of households and residents, for example the building DEBY_LOD2_52671398 with 5 households and 48 residents, resulting in differences bigger than 8 people. In this case, it was impossible to redistribute the households according to the dissertation's method. The building was then addressed as incomplete, having no given number of households and residents, and being redirected to the calculation method for case 2 in 4.2.2.

Another exception to be dealt with was the negative differences between NR_{input} and $NR_{statistics}$ also due to incoherent information provided by the input CityGML file, for example, the building DEBY_LOD2_4965750 with 5 households and 1 resident. Since negative differences are not accounted for by table 4.1, this thesis expands this method by subtracting the negative difference from the biggest households because, as already mentioned, they represent the smallest share of household sizes in Munich. For example, if the negative difference is from 3 people and the biggest household is the building contains 4 residents, the subtraction of the difference to the biggest household results in 1 resident left and therefore one new 1 resident household and one less 4 resident household.

For calculating the electricity consumption later in the methodology, the area for each household is needed. Refer to the subchapter 4.2.2 for the household area calculation method.

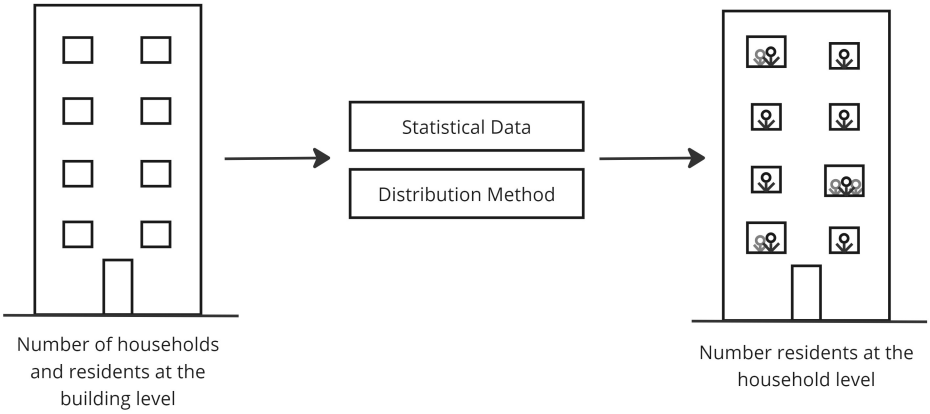


Figure 4.3: Scheme for Case 1

4.2.2 Case 2: Unknown number of households and residents in the building

Case 2 is designed to work with the majority of CityGML datasets, which do not provide specific information about the number of households and number of residents in each building. Even though Robert Kaden's dissertation provides a method for calculating the number of households and residents for buildings missing this information, this Thesis selected a method scalable to entire cities and applicable to the whole country. The method in [41] is based on the building's volume and the number of housing units per cubic meter, which was empirically determined through a site survey in Berlin Moabit for each building age class, meaning this approach is limited to Berlin's scope. In contrast, the method presented in [8] uses statistical data for the entire country. In addition, the chosen approach was tested in a major city, a mostly rural, and a suburban German county, delivering satisfactory and comprehensible results that justify its applicability.

Volume and average storey height calculation

The first step is calculating the building volume and estimating its heated area. In this Thesis, the Wall, Ground, and Roof geometries in CityGML are parsed into polygons formed by points with three-dimensional coordinates. Then, the building's 3D surface is decomposed into triangular polygons, and the volume of each polygon with respect to the origin point (0,0,0) is summed into the total building's volume. This was implemented in Python and checked against the results from the existing CG_Volume SFCGAL (Simple Features Computational Geometry Algorithms Library is a C++ library that provides advanced 2D and 3D spatial functions often used with spatial databases [48]) function.

In the following, the average floor-to-floor height should be calculated. In this thesis, this calculation was simplified by dividing the building's height by the number of floors, both

provided by the CityGML file (see equation 4.3).

$$h_g = \frac{h_m}{n_s} \tag{4.3}$$

where:

- h_g - floor-to-floor height;
- h_m - measured height from CityGML building;
- n_s - number of storeys.

As next the building's heated area is derived from the Volume through equation 4.4 when h_g lies between 2.5 - 3.0 meters and through equation 4.5 in other cases. Both equations are defined by Germany's 2013 Energy Saving Ordinance (EnEV) [49]. They determine that service and circulation areas such as entrance areas, stairwells, elevators, and corridors are heated, while technical areas such as operating rooms, cellars, and unheated attics are not.

$$A_h = 0.32\frac{1}{m} \cdot V_h m^3 \tag{4.4}$$

where:

- A_h - building's heated area;
- V_h - building's volume.

$$A_h = (\frac{1}{h_g} - 0.04\frac{1}{m}) \cdot V_h m^3 \tag{4.5}$$

where:

- A_h - building's heated area;
- h_g - floor-to-floor height;
- V_h - building's volume.

Although [8] assumes that buildings with a footprint smaller than 9 m² are non-heated buildings, to address false classifications of spaces such as garages or balconies, which should not be included in the estimation of the number of households and occupants, due to the scope of this thesis, this step is not performed. Here, the number of households and residents is calculated based on the exterior geometry of the residential buildings, and the building's function is not changed.

Estimation of the number of households per building

The German Building Typology classifies residential buildings as single-family house SFH, terraced house TH, multi-family house MFH, apartment block AP and high-rise building HRB[50]. An SFH only contains one household, a TR can be either an SFH or an MFH depending on its size,a AB must have between 3 to 12 households, and a HRB must exceed 22m height [50]. The number of households is calculated by dividing the A_h by the average household areas provided by Destatis [51] for each type of house (see equations 4.6, 4.7, 4.8). All calculation results are rounded to integers.

$$NH_{MFH} = \frac{A_h}{80.2m^2} \tag{4.6}$$

$$NH_{AB} = \frac{A_h}{62.4m^2} \tag{4.7}$$

$$NH_{HRB} = \frac{A_h}{54.3m^2} \tag{4.8}$$

where:

- NH - number of households in each type of building;
- A_h - building's heated area;

Distribution of heated areas per household

First, for accounting for the service, circulation, and structural areas, the heated area is reduced by 41%. With the number of households and the total heated area, it is possible to distribute the area to the households. This is done through a greedy algorithm [52] that determines the household area based on statistical frequency density distribution (FDD) data from [53] (see Table 4.3) and also checks if the total A_h is not exceeded and if the remaining area provides at least 40m² for each remaining household. The household areas are calculated based on the equation 4.9.

household area in m²	%	household area in m²	%
<40	5.0	120-139	10.7
40-59	17.5	140-159	6.1
60-79	23.5	160-179	2.9
80-99	17.3	180-199	1.8
100-119	12.4	200-219	2.8

Table 4.3: Frequency density distribution for household areas [53]

$$A_{household} = A_{min} + i \cdot 20 + random(0-20), \quad A_{min} = 20 \tag{4.9}$$

where:

- $Ahousehold$ - area of a household;
- $Amin$ - minimum area of a household;
- i - random index according to the statistical distribution; i = (0,...,9)

For the pseudo code refer to the paper "Determination of Household Area and Number of Occupants for Residential Buildings Based on Census Data and 3D CityGML Building Models for Entire Municipalities in Germany". For the code implemented in Python for this Thesis refer to the GitHub Repository [46].

Determination of the number of residents per household

The estimation of the number of residents per household is similar to the previous estimation of household areas. A greedy algorithm determines the number of occupants based on an FDD also from Destatis [53]. This statistical data can be seen in table 4.4 and describes the probability of a specific occupancy depending on the household area.

household area [m²]	1 P. [%]	2 P. [%]	3 P. [%]	4 P. [%]	5 P. [%]	6-8 P. [%]
<40	89.4	9.1	1.5	0	0	0
40-59	69.9	24.1	4.5	1.5	0	0
60-79	42.1	38.3	12.6	5.5	1.5	0
80-99	28.4	39.1	17.2	10.6	3.2	1.5
100-119	20.4	38.9	20.0	14.7	4.1	1.9
120-139	15.3	35.7	21.8	19.3	5.6	2.3
140-159	12.9	33.2	21.8	21.9	7.2	3.0
160-179	11.7	30.5	21.6	23.7	8.6	3.9
180-199	11.2	29.3	21.2	23.9	9.7	4.7
>200	11.1	27.2	19.9	23.4	11.2	7.2

Table 4.4: Frequency density distribution for number of occupants depending on the size of the household [53]

For the pseudo code refer to the paper "Determination of Household Area and Number of Occupants for Residential Buildings Based on Census Data and 3D CityGML Building Models for Entire Municipalities in Germany". For the code implemented in Python for this Thesis refer to the GitHub Repository [46].

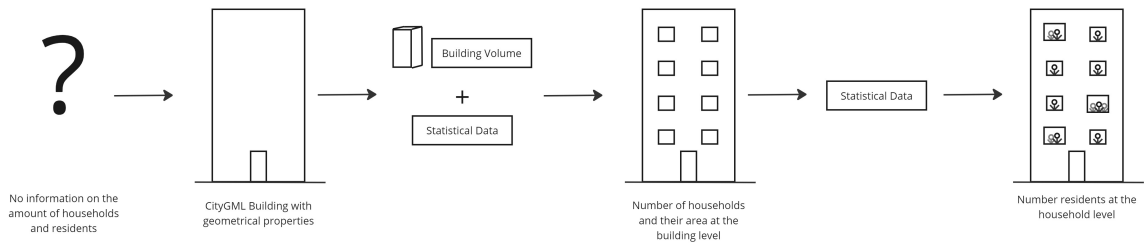


Figure 4.4: Scheme for Case 2

As mentioned before in subchapter 2.2.1, starting in the CityGML version 3.0, households in the building elements should be assigned as "building units". Nevertheless, this thesis works with a dataset in version 2.0, and as a solution, the simulated households were saved in CityGML as "building parts" and the number of residents and household area as "generic attributes". However, it is essential to mention that this is a mishandling of the attribute "building parts".

4.3 Yearly electricity consumption estimation using an LSTM Network

The success of LSTMs briefly mentioned in the chapters 1.1 and 2.3.4 for time series forecasting lies in its ability to recognize blocks of sequenced data. The amount of research

on this topic shows its success in forecasting electricity consumption. For these reasons, an LSTM network was developed to estimate the annual electricity consumption curves for the simulated households for the CityGML buildings. For building the network, this thesis was inspired by [9] and followed these adapted steps:

1. Data collection and Preparation
2. Determination of the network architecture.
3. Determination of the training parameters.
4. Implementation.

The following figure 4.5 represents an overview of the final LSTM Network.

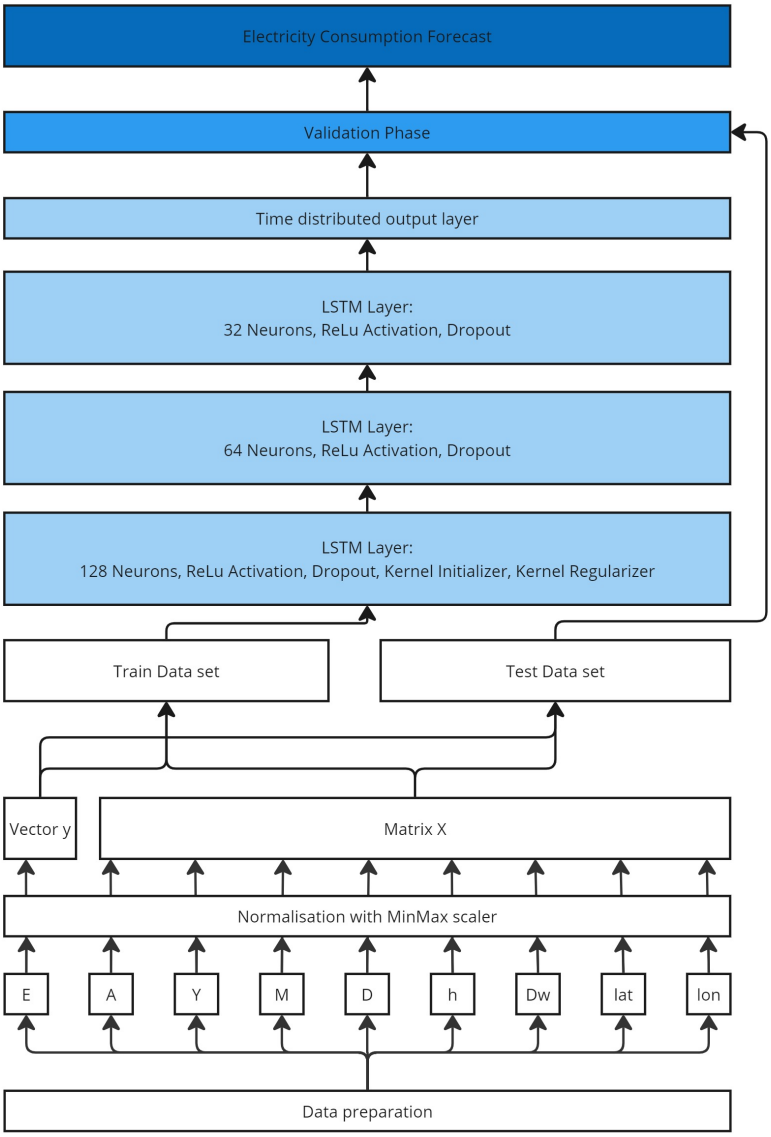


Figure 4.5: Overview of the LSTM Framework

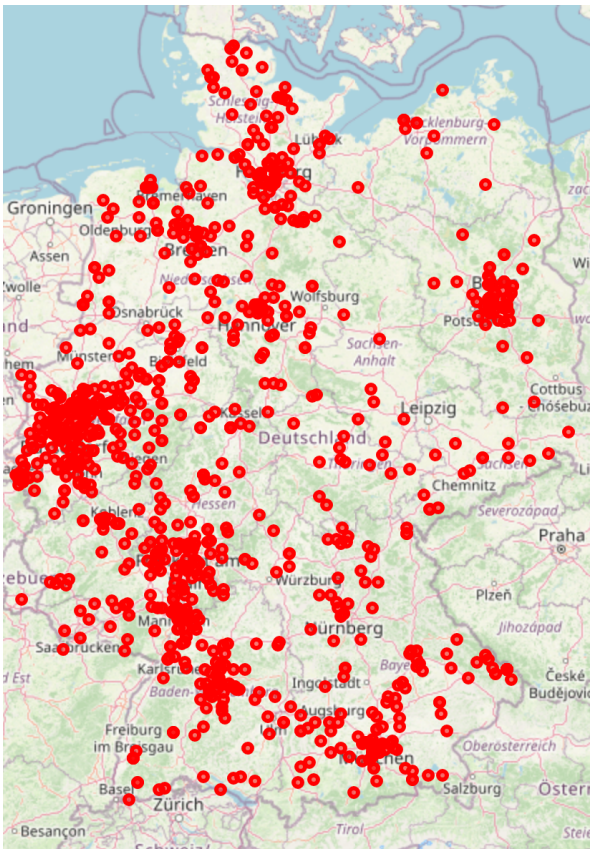


Figure 4.6: Map of Germany with marked location of the electricity measuring sensors

4.3.1 Data collection and Preparation

A challenge when using data-driven methods such as ANNs is to find a suitable dataset. This thesis utilized the data provided by Open Meter DE, a platform for energy-related measured data with over 2500 sensors such as electricity, gas, and water meters, aiming to contribute to the energy transition by helping network operators, research, and service providers.

The appropriate data set for this thesis was filtered by the following categories: energy type, measurement category, object, and object use, where the values selected were respectively electricity, consumption, private, and household. The measured data period is different for every sensor, where you can have one sensor with data from November 2017 to August 2024 and another from July 2020 to August 2024. The frequency of the measured data is also different depending on the sensor, but most of them were measured every 15 minutes. For these categories, a total of 1094 sensors were placed in households with areas ranging mostly from 30 to 375 m^2 with only 17 sensors with bigger areas and in cities in entire Germany (see figures 4.6).

In this thesis we tried to work with all sensors using hourly values from 2021 to 2023, including blocks of two or three years in a row for capturing consumption characteristics over the years. The amount of data was too big for the computer processing capacity. Only joining every sensor data to one single file, comprised of more than 20 million measurement lines, took around 13 hours. Training the LSTM with so much data required too long and therefore interfered with the network-building process. Because of this, the selected sensors were reduced to the city of Munich. A total of 13 sensors were downloaded for the time period

from 01.01.2021 to 31.12.2023, with hourly values. Hourly values were chosen to shrink the amount of data to be processed and, therefore, to improve the processing speed. Since some sensors had data for the whole time period and others for only one of the years, this resulted in 29 samples of 8760 measurements ($365 days \cdot 24 hours$).

Open Meter DE id	Postal Code	Latitude	Longitude	Household Area [m²]
3f9eccea-aa85-4987-9fa4-ccdc85e92b9e	81549	48.091676	11.593581	63
7840dd62-042f-42e8-9179-87909fc6b17e	81245	48.155400	11.427546	75
5c059e6d-f3a1-493a-bc33-c288129833cd	81829	48.134371	11.663611	75
bd00a4b7-d6a3-4310-8173-f16e868e96c2	81669	48.120738	11.605218	78
db2456e5-357e-4f35-97cf-2072ffa52c02	81667	48.128127	11.600001	98
946e85d5-1700-4de4-aa43-80d72defebfe	81245	48.155400	11.427546	110
33b6957c-d968-4397-bec1-4ead226b5f68	80689	48.128925	11.494575	115
60b63962-cc0e-4cea-adcd-6c0308c2cb1c	80687	48.140593	11.512824	120
6f6730f9-83d4-45eb-bd1c-1df7dea9b78e	80993	48.184570	11.505790	151
b68504d5-b0d7-44af-a129-745dcc7f4345	81243	48.149720	11.439522	180
0ceb32a7-a1ab-4a67-9d24-af9bdae31c11	81476	48.098454	11.503066	200
12c77d0d-c6a0-4fc8-ae44-022c11e38aa3	81249	48.150683	11.413140	300
b787342d-b5a3-44c4-bb5a-1350f6b614dd	81829	48.134371	11.663611	462

Table 4.5: Information table about electricity measuring sensors in Munich

The relevant variables for this network are the time steps, the geographical location of the sensors, the area of the households with the installed sensors, and the target variable is the electricity consumption. These variables were chosen because they allow the network to predict the electricity consumption for the CityGML households, for which the area was estimated, and the geographical location is given in the CityGML file. The first step is to prepare them for the network, transforming some features into numerical values since it is the only data type accepted for fitting a model. This methodology chose to work with the postcode since it provides a more exact household location than the information city and also for working only with sensors within Munich. The latitude and longitude were not given by the OpenMeter DE Platform but converted from the given postcode in Python because, contrary to the postcode itself, they follow a logic structure that could teach the LSTM to capture spatial features. The measurement time steps are provided in DateTime format YYYY-MM-DD hh:mm:ss; this was separated into year, month, day, and hour, so every feature is a number. The feature day of the week was also added with the Python Pandas Package "dayofweek" function that assigns numerical values from Monday, a zero, to Sunday, a six.

The features used for the LSTM can be described as follow:

1. Vector containing the sequence of electricity consumption of all samples $E = e_i, ..., e_{254040}$

2. Vector containing the household area of a specific sensor A, range = (63,...,462)

3. Vector containing the year of each measurement Y, range = (2021, 2022, 2023)

4. Vector containing the month of each measurement M, range = (1,...,12)

5. Vector containing the day of each measurement D, range= (1,...,31)

6. Vector containing the hour of each measurement h, range= (1,...,24)

7. Vector containing the days of the week D_w, range = (0,...,6)

8. Vector containing the latitude coordinate of the household lat, range= (47°-55° N)

9. Vector containing the longitude coordinate of the household lon, range= (5°-16° E)

The vectors A, lat, and lon hold the same value for every sample because those features do not change over time.

To comprehend with what type of data the Network will be learning from, it is essential to understand the basic statistical properties from the target value "Electricity Consumption". The data contains a mean value of 0.468903 kWh and a standard deviation of 0.735101, indicating a spread of the data points in relation to the mean, a minimum value of 0.0 kWh, and a maximum value of 14.35 kWh. Also, the quartiles (a type of quantile which are cut points that divide the data into equal probability distributed parts) indicate that 25% of the data lie under 0.17, 50% lie under 0.3 and 75% lie under 0.49. This information can be seen in figure 4.7, which plots a highly skewed distribution of the target value, data that is not symmetrically distributed around the mean value. No Nan (not a number) values were found in this dataset.

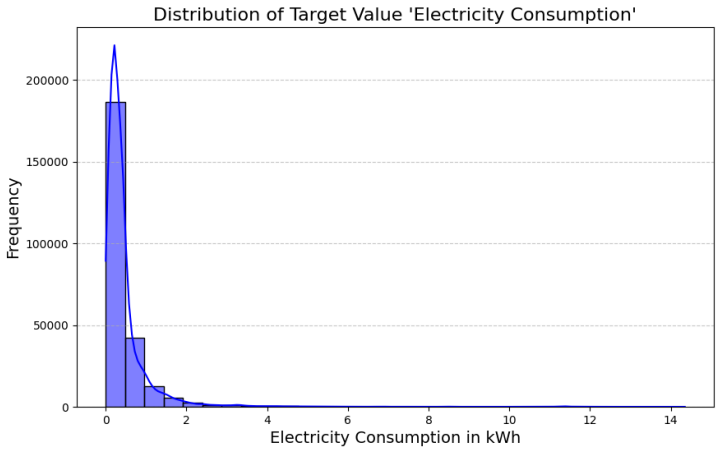


Figure 4.7: Histogram of the original data for Electricity Consumption

It is also necessary to scale the data into ranges that are better for the network to interpret because the activation functions are sensitive to the scale of input data. Sometimes unscaled data can slow down the learning and convergence of the network or even prevent it from learning the problem [54]. The data was normalized using the MinMaxScaler function from the Keras Python package, which scales the data into a given range; here, the default range 0 to 1 is used. For the feature area, the scaler was fitted into a randomly created vector with values ranging from 20 to 500 m² because the original vector ranged from 63 to 462 m ², and for areas under the original boundary, for example, 40 m², the prediction of the network resulted negative.

For the prepared input, the vectors are concatenated into a matrix $X = \{\mathbf{A}^\top \mathbf{Y}^\top \mathbf{M}^\top \mathbf{D}^\top \mathbf{h}^\top \mathbf{D_w}^\top \mathbf{lat}^\top \mathbf{lon}^\top\}$ and the target value is transposed to vector $y = \{\mathbf{E}^\top\}$. As mentioned before, the matrix consists of 29 samples, each with 8760 lines of data. This was separated into 24 samples for training the network and 5 for testing the network's performance. The pictures in figures

A.1a to A.15a in Appendix A plot the 29 training samples.

4.3.2 Network's architecture

After the data preparation, the next step is to define the network's architecture by specifying the number of layers, their units, and the network's components. This will determine how the network should operate when receiving inputs, processing them, and outputting results.

During the training phase, a graph of the loss function in each epoch was plotted for each architecture to monitor if the loss decreases over epochs, which means the model is learning. An ideal loss over epoch graph should show the train and validation loss decreasing and stabilizing around the same point [54]. To find the best architecture, 10 different structures were tested (see Table 4.6). The Mean Square Error (MSE), defined as the average of the squared differences between the predicted and actual values, was assigned as the loss function to evaluate the training phase. For evaluating the network's performance, the Root Mean Squared Error was used as a metric. This metric is defined as the square root of the MSE, measuring the average magnitude of the prediction errors in the same unit as the target variable. By taking the square root, RMSE emphasizes larger errors more than smaller ones. To find the best architecture, the models were tested using the function model.evaluate from the Python Keras package and a batch size of 16, a learning rate of 0.0001, and 15 batches. These hyperparameters were later adapted to the best combination.

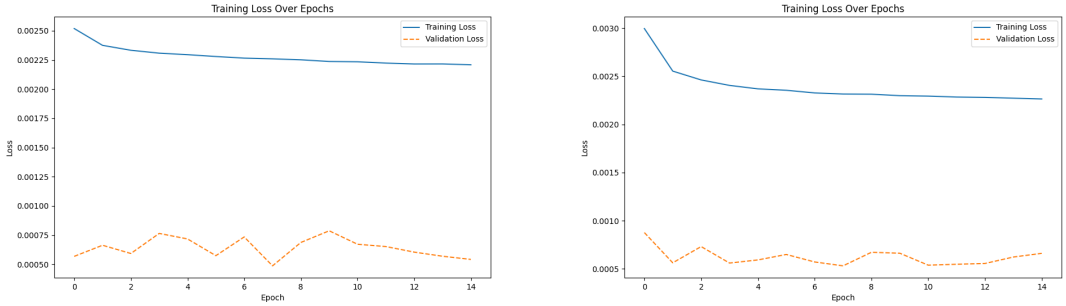
The building process started with simple models containing 1, 2, and 3 LSTM layers. As explained in 2.3.2, a layer is a collection of neurons that process the input data and output a result to the next layer by applying its weights and activation function. The first layer receives the raw data and the layers in between are called "Hidden Layers" as seen in figure 2.5a. The amount of neurons of each layer can be seen in table 4.6. The layers used the Adam optimizer and the ReLU activation function, the most successful and widely implemented activation function [55]. The learning curve, which plots the loss over the epochs, for these models indicated overfitting because the curve was ideal for the training phase and bad for the validation phase, which means that the model performed well on the training data, but not on test data. Therefore, the model required further improvement. The loss over epoch curve for the LSTM Model with 3-layers (3L model in table 4.6) can be seen in Figure 4.8a.

Techniques to avoid overfitting were applied to achieve a better generalization, a characteristic that defines Networks that can perform well in test data. The following methods were gradually applied to the 3L LSTM since it performed better than the 1 and 2-layer Networks. First, a Dropout of 0.3 was added to each layer, deactivating randomly 30% of the neurons in each training step by setting them to zero and forcing them not to participate in the backpropagation process. This improved the RMSE of the simple LSTM Networks but still delivered an overfitting model. Adding a L2 kernel regularization of 0.01 stopped the overfitting and delivered a satisfactory validation curve (see Figure 4.9a). This technique forces regularization by controlling the weights generated by the backpropagation process and penalizing large weights proportionally to the sum of the squared weights [56].

After this step, the model was satisfactory, but further potential improvement techniques were tested. Adding gradient clipping, a method to prevent the norm of the gradients exceeding 1.0 during backpropagation to avoid exploding gradients (accumulation of gradients resulting in a very large gradient that can disable the network as mentioned before in sub-

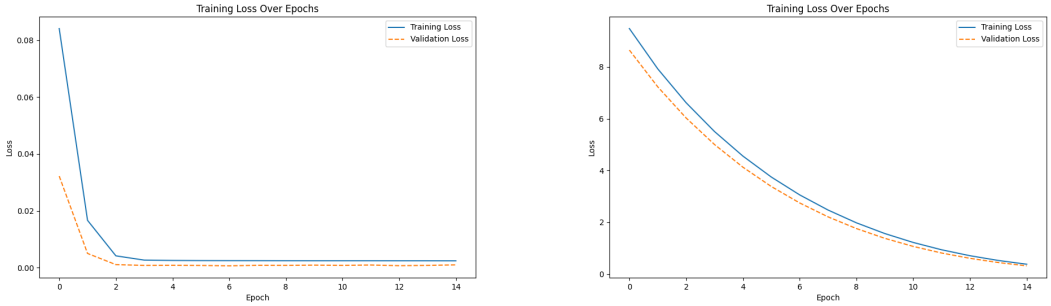
chapter 2.3.4), improved the MSE score further. In addition, a TimeDistributed output layer was added, which also improved the RMSE score. This type of output layer generates predictions for each time-step, while normal output layers generate one prediction for the entire input sequence. This type of layer is essential for time-series forecasting to preserve the temporal structure of the data [56]. Last, a He Normal Kernel initialization was added because it is an initialization type optimized for the ReLU activation function [57]. This technique initializes the weights in a controlled way by paying attention to the previous layer to find a global minimum of the loss function faster and more efficiently. This delivered the best combination of RMSE score and Learning curve of all architectures, and it was consequently chosen as the final architecture.

The comparison of the model's performance can be seen in figure 4.10. Even with the same network's structure, every time a model is generated in Python, the values of the metrics slightly alter due to the random initialization of weights. This makes the network converge to different local minima across runs. Therefore, the final RMSE values shown in figure 4.10 are the mean of the RMSE resulting from ten runs of each model.



(a) 3L **(b)** 3LD

Figure 4.8: Learning curve for different overfitting models



(a) 3LD_KR **(b)** 3LD_KR_GC_TD_KI

Figure 4.9: Learning curve for different models

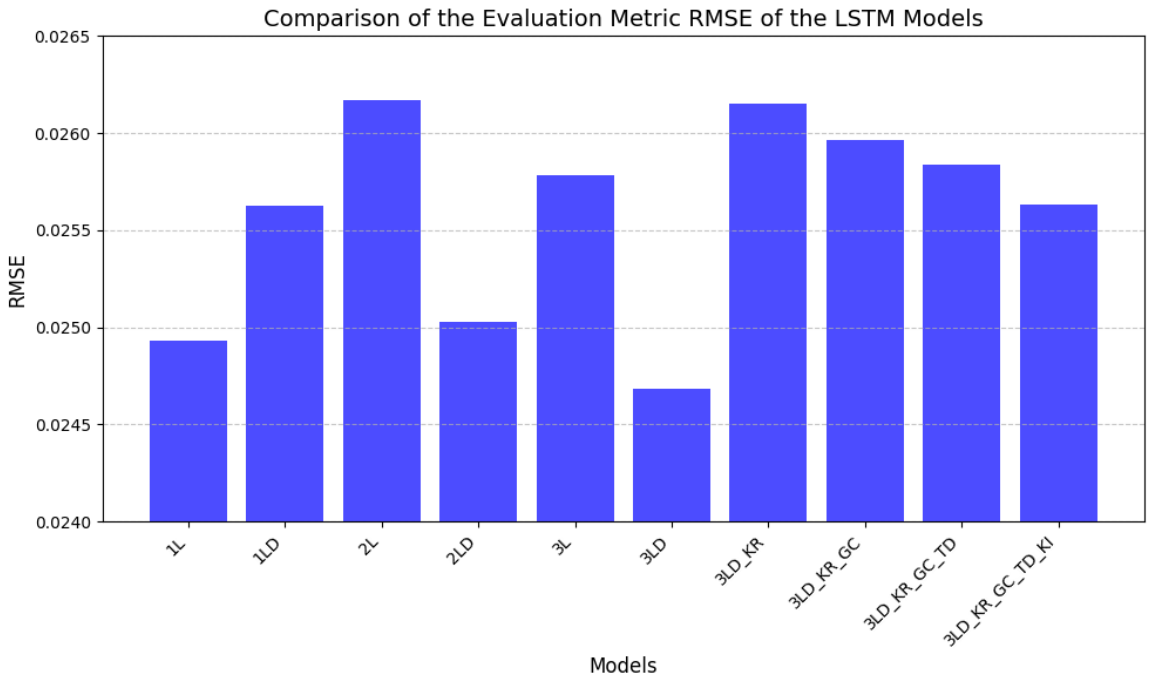


Figure 4.10: Comparison of the RMSE for each Networks Architecture

Table 4.6: Information Table for Different LSTM Models

Model (Short)	Model Description	Number of neurons in each layer	RMSE Value
1L	LSTM 1 layer	64	0.024933
1LD	LSTM 1 layer with Dropout	64	0.025625
2L	LSTM 2 layers	64, 32	0.026169
2LD	LSTM 2 layers with Dropout	64, 32	0.025028
3L	LSTM 3 layers	128, 64, 32	0.025784
3LD	LSTM 3 layers with Dropout	128, 64, 32	0.024683
3LD_KR	LSTM 3 layers with Dropout and Kernel Regularization	128, 64, 32	0.026153
3LD_KR_GC	LSTM 3 layers with Dropout, Kernel Regularization, and Gradient Clipping	128, 64, 32	0.025964
3LD_KR_GC_TD	LSTM 3 layers with Dropout, Kernel Regularization, Gradient Clipping, and TimeDistributed Layer	128, 64, 32	0.025837
3LD_KR_GC_TD_KI	LSTM 3 layers with Dropout, Kernel Regularization, Gradient Clipping, TimeDistributed Layer, and Kernel Initialization	128, 64, 32	0.025634

4.3.3 Determination of the training parameters

As mentioned before in subchapter 2.3.3, choosing optimal values for hyperparameters is difficult, so they are often chosen based on literature recommendations, experience, trial and error, or alternatively through tuning strategies [31]. Therefore, the parameters learning rate, batch size, and number of epochs were adjusted by trial and error for the best network architecture in the previous subchapter. Experiences with the learning rates, epochs, and batch sizes in 4.3.3 were conducted.

As explained in the subchapter 2.3.3, the batch size represents the number of samples processed before computing the gradient and updating the weights. Therefore, the values were chosen to represent a certain number of samples. Since each sample in this case is an hourly time series for a year and accounts for 8760 lines in the Matrix X, the values in 4.3.3 are for 1, 2, 4, 8, and 16 samples. Compared to the test data, the best consumption curves were found using a learning rate of 0.01, 20 epochs, and a batch size of 8 samples (70080).

1. $Learningrates = [0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005, 0.00001]$
2. $Epochs = [10, 15, 20, 25, 30]$
3. $Batchsizes = [8760, 17520, 35040, 70080, 140160]$

4.3.4 Implemetation

Finally, the model of the LSTM Network is defined and can be used for estimating the electricity consumption for the modeled households in the CityGML case study Harthof dataset. Here it is necessary to create an X matrix for the model to predict the target value y.

For each household, the Matrix X contains 8784 lines accounting for every hour of the leap year 2024. The columns of the matrix are defined as explained in 4.3.1. The latitude and longitude of each household were calculated through the coordinate transformation of the easting and northing coordinates of each building provided by the CityGML file with the transformer.from_crs function from the pyproj Python package. The area of each household was calculated with the method presented in subchapter 4.2.2. The vectors containing the information about the date, hour, and day of the week were automatically generated in Python for the year 2024. The prediction year is, therefore, easily adjustable in the code.

For performing the prediction, a loop for every 24 hours of the year calls the function model.predict from the Keras Python package and outputs the next 24 hours estimated from the developed LSTM model. Every day is appended to form a year of predictions and saved as a CSV file directly in the visualization folder. The sum of the hourly values for the year of 2024 were used as input for the BDEW H0 SLP calculation for later comparison.

4.4 Yearly electricity consumption estimation using a standard load profile

Even though the standard load profile for household H0 defined by BDEW has limitations as mentioned in 1.13.1, it was chosen for this thesis to compare the LSTM-generated energy consumption to the H0 SLP-generated consumption because it is the official method utilized by the City Utilities.

For each simulated household in the Harthof CityGML file, two yearly electricity consumption curves were generated using the BDEW method. As mentioned before in 3.1, the profiles are scaled based on annual values. Since no real value for the yearly power consumption for these simulated households is known, two options for calculating them were conducted. First, the number of residents estimated in the subchapters 4.2.1 and 4.2.2 are used to define the annual electricity consumption using statistical data. Second, the LSTM-generated values were summed to deliver a yearly value.

For estimating the yearly electricity consumption value for each household based on the number of residents calculated in the subchapter 4.2.1 and 4.2.2, so it can be used as input for scaling the H0 SLP, statistical data from BDEW in 2021 in [58] was used. The report on the average electricity consumption (excluding heating electricity) per household by household size determined that households from sizes one to five or more people consumed respectively 1900, 2890, 3720, 4085, and 5430 kWh per year. These values were assigned as the attribute ELC_{stats} (statistically-generated yearly electricity consumption) to the CityGML households based on the number of residents defined in 4.2 and used as input for the standard load profile calculation. On the other hand, the sum of the values generated by the LSTM were also saved as attributes of the CityGML households as ELC_{LSTM} (LSTM-generated yearly electricity consumption) and used to calculate a third consumption curve. For comparing this curve to the other two and to the smart meter data, the number of residents for the households in the test data samples was calculated using the method in 4.2.2.

For implementing the H0 SLP, a Python library called demandlib can be used for generating the load profiles. The profile generated from the real consumption data from the 70s is saved in the folder bdew_data in the file selp_series.csv. First, a profile is generated through the Python Class ElecSlp and manually defining the holidays for the desired year. The profiles generated by this class are automatically dynamized through the function in [39] to deliver a more harmonized curve. Then, the function get_profile scales the profile, which till here was still normalized to a yearly consumption of 1000 kWh, based on the given annual consumption.

Chapter 5

Visualization

This thesis utilizes 3D city models to simulate the electricity consumption of each apartment in each building. It is important to visualize the simulation results using the geographical and geometrical properties of the CityGML data. Therefore, the 3D Web Map Client package from the 3DCity Database, a database for sorting and managing 3D city models efficiently, was chosen for this task [59]. The Web Map client is a web-based high-performance tool for visualizing and exploring large semantic 3D city models, developed with Cesium Virtual Globe, an open-source JavaScript library. This tool was chosen due to its open-source character, which allowed an extension of the JavaScript code to show the results of this Thesis. In addition, the tool already allows the semantic data of the buildings to be shown as tables. The tool requires a browser, which can be Safari, Firefox, Chrome or Opera, and the installation of Node.js, a software that allows running JavaScript outside of a web browser.

The first step for visualizing the 3D CityGMl file in the Web Map Client is to convert the GML file into a Collada file or Cesium 3D Tiles. For converting into a Collada file, it is possible to use the tool Importer/Exporter from 3D CityDB by importing the file into the database and then exporting it as a Collada file. For this, the step-by-step is explained in the official documentation [60]. It is possible to use the Cesium ion website to convert to Cesium tiles by uploading your dataset and exporting it as Cesium Tiles.

The semantic data is shown by connecting a Google Spreadsheet with the web map client through its public sharing link. This thesis generates the table automatically when running the methodology's main script. Nevertheless, this table needs to be manually added to a Google spreadsheet. This is the semantic information showing function already included in the web map client, which shows semantic information for the building. However, this thesis simulated the households and their number of residents, area, the electricity consumption curve, and the total yearly consumption value for 2024. Therefore, there should be a table that shows this information at the household level. The JavaScript script.js function inside the folder 3dwebclient was edited to achieve this. A button "Show Details" in the row building_parts was added to the main semantic table, that when clicked shows the table containing the semantic information such as ID, area, number of inhabitants, yearly electricity consumption, for each household in the previously selected building. For each household a button for plotting the load profile was placed in the extreme right of the table. After clicking, the "Load Profile" button generates a pop up window for the user to select for which day or month of the year 2024 or for the entire year of 2024 to plot the graph. The BDEW-generated and the LSTM-generated curves are plotted for comparison and the window containing the plots also holds a button for exporting the data only for the selected time period. The changed script.js file can be found in the GitHub repository[46] for this Thesis.

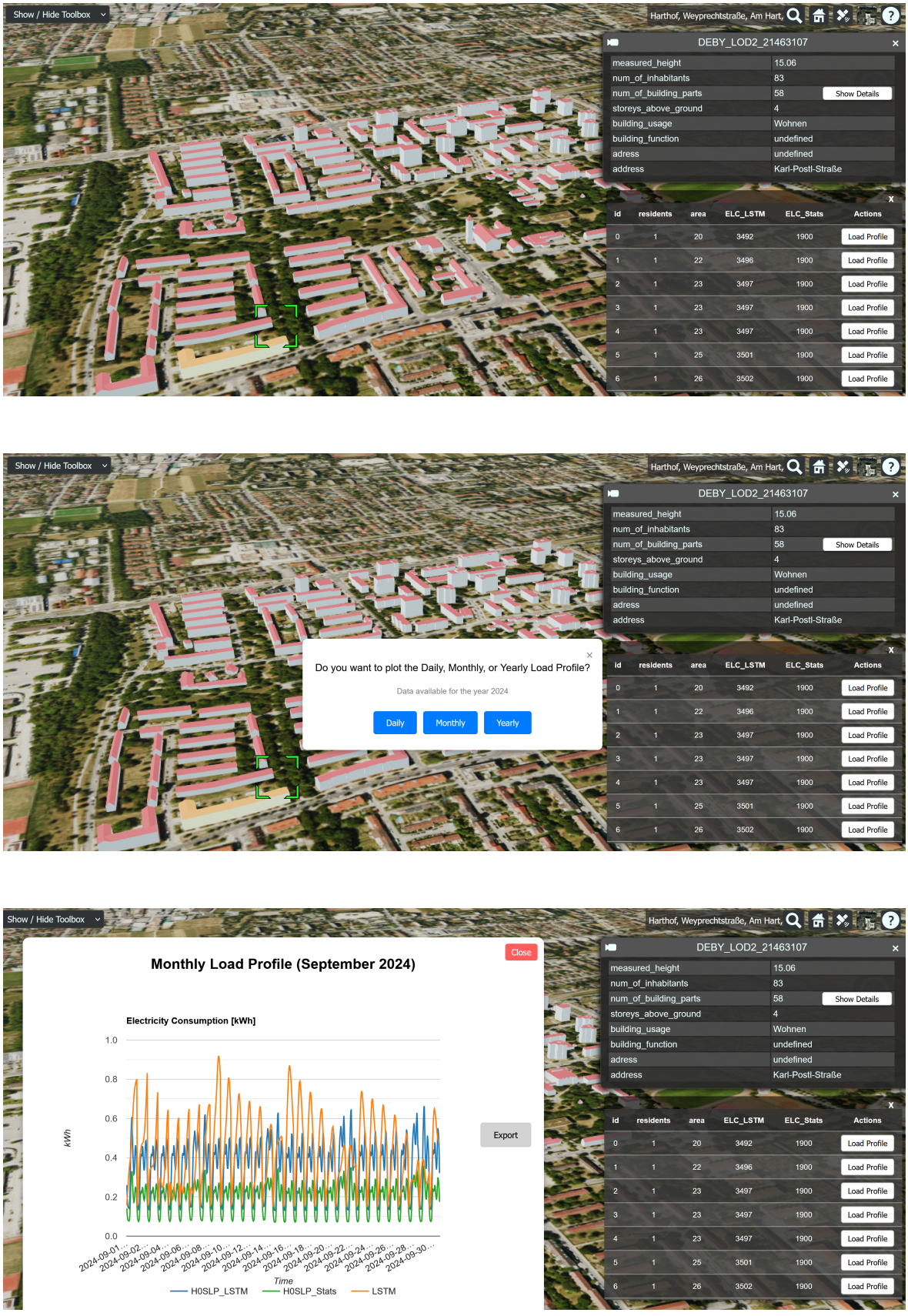


Figure 5.1: Visualisation of the building and households semantic tables and the daily consumption plot

Chapter 6

Discussion and Evaluation

This thesis presents different ways of modeling electricity consumption for households in the most accurate way possible. 3D City Models helped as backbones for the calculation of the electricity consumption curves as well as for storing household data. Robert Kaden's [41] provided a method for redistributing the residents to the household level for the CityGML buildings that already contained information about the number of households and residents at the building level. Köhler et al. [8] provided a method for calculating the number of households and their residents from CityGML buildings from scratch based on statistical data. This work successfully put those methods into practice. To complete the information at the household level, this work predicted the yearly electricity consumption curve for households through three different approaches. Furthermore, the application 3dweb map client was adapted to visualize the results with the 3D city model.

The research done for the thesis showed that load profiles can be derived from bottom-up or top-down strategies. Here, the electricity consumption at the household level was calculated based on the number of residents, area, and geographical location, being characterized as a bottom-up model.

The methodology used for creating the households and their attributes is a good basis for simulations of the residential electricity consumption in Germany due to the adaptability that the CityGML Standard provides. This semantic 3D city model shows that the number of households and their residents can be derived from the geometrical properties of the buildings combined with statistical data. It also allows the further aggregation of the values calculated in this thesis, such as the area, the number of residents, and the yearly electricity consumption at the household level, to bigger dimensions with an available CityGML model, such as building, city block, or city district level.

The LSTM was logically built by seeking an optimal learning curve and the lowest RMSE possible for the model by adding techniques for better generalization and finding the best combination of the learning rate, epochs, and batch size. The H0SLP curves were generated with the official BDEW method, using the sum of the LSTM or the average yearly electricity consumption based on the number of household residents statistics to scale the profile. To evaluate the electricity consumption predictions, the results were compared to the smart meter data from [45] previously separated to be test data for the network. The RMSE between the LSTM- and the H0SLP-generated results and the test data were respectively 0.325 and 0.316. The maximum deviation between the values was 5.208 and 5.162, respectively. The plots can be seen in figure 6.2. The values are very similar, with the H0SLP being slightly better. However, when visually inspecting the plotted annual consumption curves, the LSTM seems to outperform the H0SLP methods for yearly predictions as it better captures the ran-

domness of human consumption (see figure 6.3).

In addition, the model seems to relate electricity consumption to the household area, so the smaller the household, the smaller the consumption. This can be seen in figure 6.2 for samples 25 and 26, measured with the same sensor located in a household of 78 m^2, where the model seems to lack a good performance. However, the disparity of the values between the generated consumption curves and the measurements is probably attributed to the unusually high load consumption for this household size.

An inspection of the daily plots for 01.01.2024 across the five test samples (see figure 6.4) does not allow a definitive evaluation of the performance of the three predictions, as the test data apparently does not follow any pattern. While the H0SLP-generated curves demonstrate a closer alignment with the smart meter measurements for sample 26, this consistency is not observed across the remaining samples. Nevertheless, the H0SLP more accurately represents typical daily electricity consumption patterns, characterized by two distinct peaks during the day and minimal activity during nighttime hours. The Network's ability to better model yearly trends rather than daily curves could be related to the batch size, which defines that the model must go through eight yearly samples before updating its internal parameters. This probably favored long-term trends, indicating a need for further optimization of the model for short-term accuracy.

Figure 6.1 takes a closer look into the LSTM annual prediction for sample 27. This plot suggests that the model managed to capture the increasing consumption trend in the winter months and the decreasing consumption in the summer. However, the model requires further improvements because some values returned negative, which is incorrect for electricity consumption prediction. This was solved by clipping the negative values to zero. There are some possible reasons for the false generation of negative values, such as incorrect scaling, noisy and/or unrepresentative data, wrong activation functions, improper weight initialization, underfitting, and/or overfitting. The cause of the negative prediction seems to be noisy and unrepresentative data since the development of this model scaled the target value properly to values between 0 and 1, used the non-negative ReLu activation function, correctly initialized the weights with the He Normal kernel initialization, and the learning curve does not show signs of underfitting or overfitting. To understand this, samples 1, 2, 22, and 23 in the Appendix A contain measurements in the opposite direction, which the Network likely captured.

As seen in figures 4.7 and A.1a to A.15a, the training data is highly skewed and does not represent real-world patterns as it only covers 29 yearly samples from 13 households. An unrepresentative and imbalanced data set can interfere with the performance and accuracy of data-driven models such as neural networks [61]. The amount of samples is probably not enough to learn how to generalize this problem, as neural networks are normally trained with thousands of samples. Ideally, the training data would comprise smart meter data for at least one household in each size range in 4.3 for each neighborhood in Munich. On the other hand, having a bigger data set also involves having a better computational capacity than what is available for this thesis. To improve the model's prediction capacity, future analysis of the model and improving the training data set are encouraged. Also, it is suggested to train the model with better computational capacity using the smart meter data for the entire country available in [45].

The results suggest that the model captures the connection between electricity consumption and household size, where smaller homes generally use less energy. This pattern is

visible in Figure 6.2 for samples 27 to 29, recorded in a 98m² household. Here, the model's predictions better match the dimension of the smart meter data, as the actual consumption is reasonable for a home of this size. However, the model seems to perform poorly for samples 25 and 26, located in a 78m² household. This discrepancy is likely not due to model limitations but rather to unusually high energy consumption in that household, which deviates from typical patterns.

The negative aspects of the results may be attributed to the choice of the loss function, which may not represent the real-world goals of the model. Employing a time-series-specific loss function, such as Dynamic Time Warping (DTW), could improve the model's ability to capture the timing and shape of the consumption patterns, leading to better results. Integrating an attention mechanism could also benefit the LSTM model, enabling it to better identify and concentrate on critical time steps, particularly during periods characterized by rapid fluctuations or significant peaks.

A positive characteristic of using the LSTM is its flexibility for adding and excluding new parameters from the X matrix. Similar to [9], further research should explore the addition of the temperature curves in the cities where the households are located. This thesis could not provide this due to the costs of using the OpenWeatherMap API (the student license was requested, but no response was given). Furthermore, classifying the days as school vacations, as mentioned in [14], can also be an advantage since many families go on trips, a habit that affects the consumption curve. However, automating this type of day is complex due to its yearly changes. On the other hand, the option of excluding parameters should also be explored to see if the model's generalization is improved. For example, reducing the two coordinate features to a single space-filling value [62] can reduce the Network's complexity without losing the spatial information, which can positively affect the model's efficiency.

Generating the residential standard load profile from BDEW, the H0SLP, is limited since it requires knowing how much electricity the household consumes per year. This information is personal and, therefore, not available when performing a simulation for multiple buildings. By calculating a possible yearly consumption value for each household based on the LSTM values or the number of residents, this thesis enabled the use of the BDEW method for simulation purposes when the real consumption value is unknown.

For the H0SLP-generated consumption curve that was scaled on the yearly average electricity consumption based on the number of household residents from [58], there is also room for improving the annual consumption values. Sources, such as the energy company Vattenfall or the ImmoScout website, suggest the formula 6.1 for calculating the yearly electricity consumption for households based on their number of residents, their area, and the number of large household electric devices. This thesis works with five values for ELC_{stats}, but the formula would deliver more personalized values. The first two factors were calculated in this thesis. On the other hand, information on the number of devices per household size is not available.

$$ELC = NR * 200kWh + NED * 200kWh + A_{household} * 9kWh \tag{6.1}$$

where:

- ELC - total yearly electricity consumption;
- NR - number of residents in the household;
- NED - number of large household electric devices;
- $Ahousehold$ - area of the household;

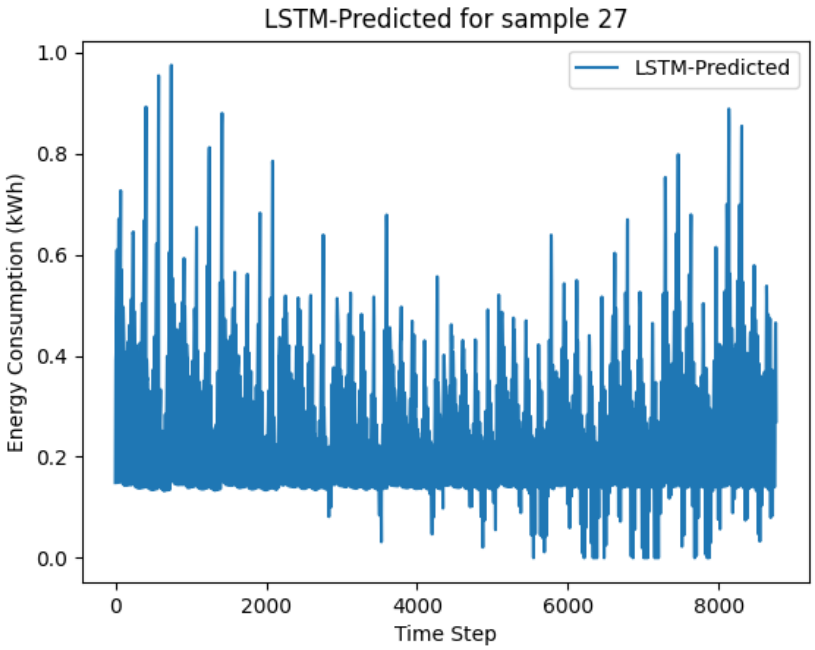
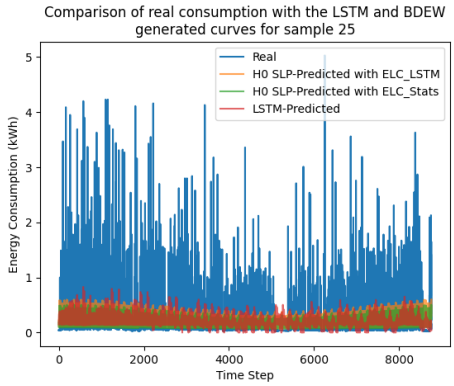
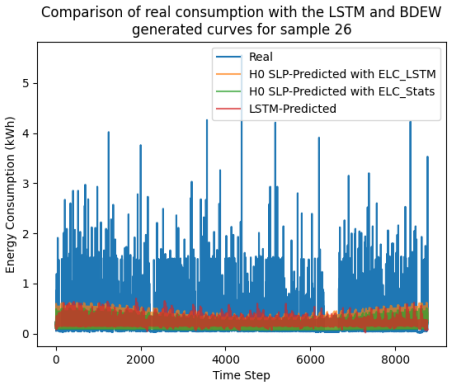


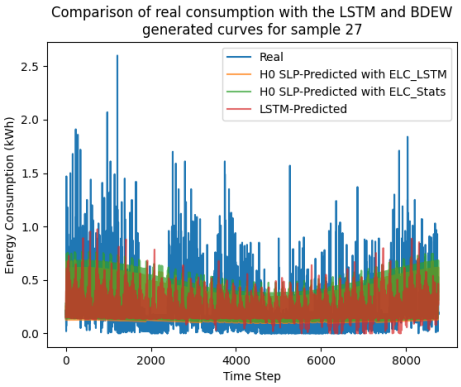
Figure 6.1: LSTM predicton for sample 27



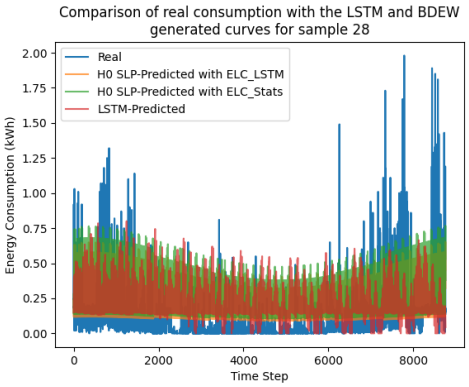
(a) Sample 25



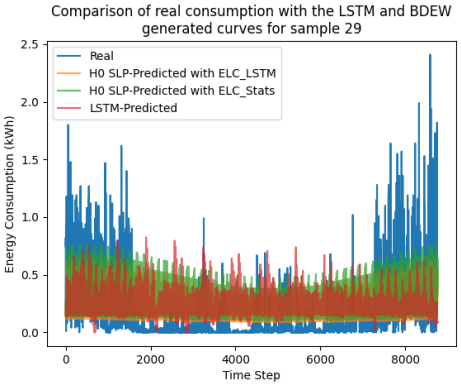
(b) Sample 26



(c) Sample 27

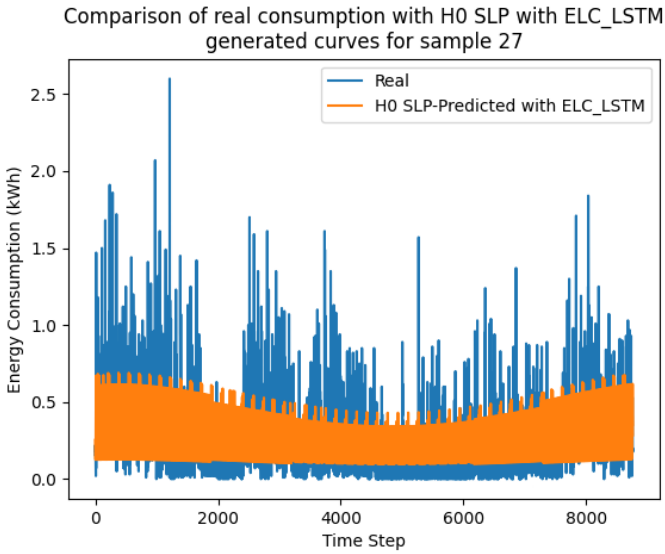


(d) Sample 28

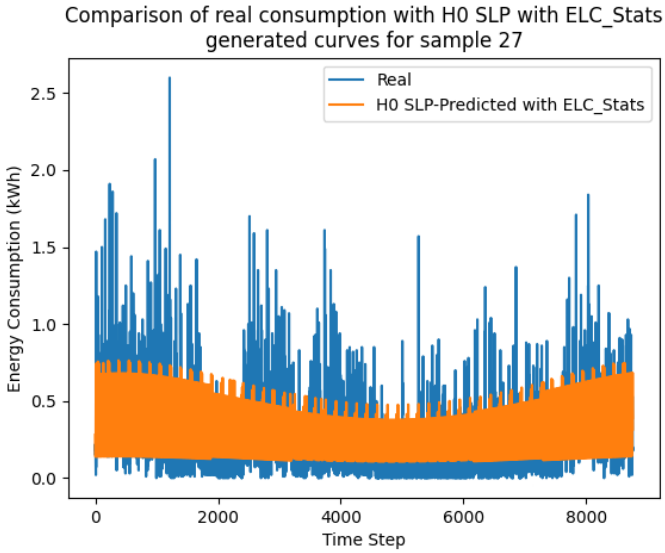


(e) Sample 29

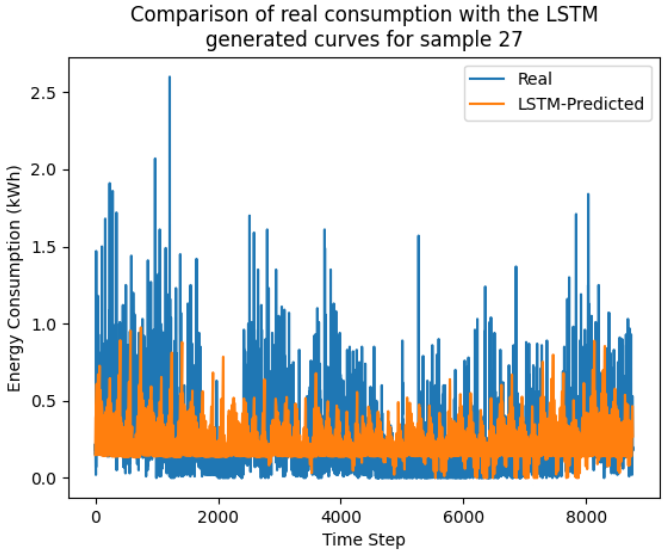
Figure 6.2: Comparison of the electricity consumption curves



(a) H0SLP-generated curve scaled with sum of LSTM values

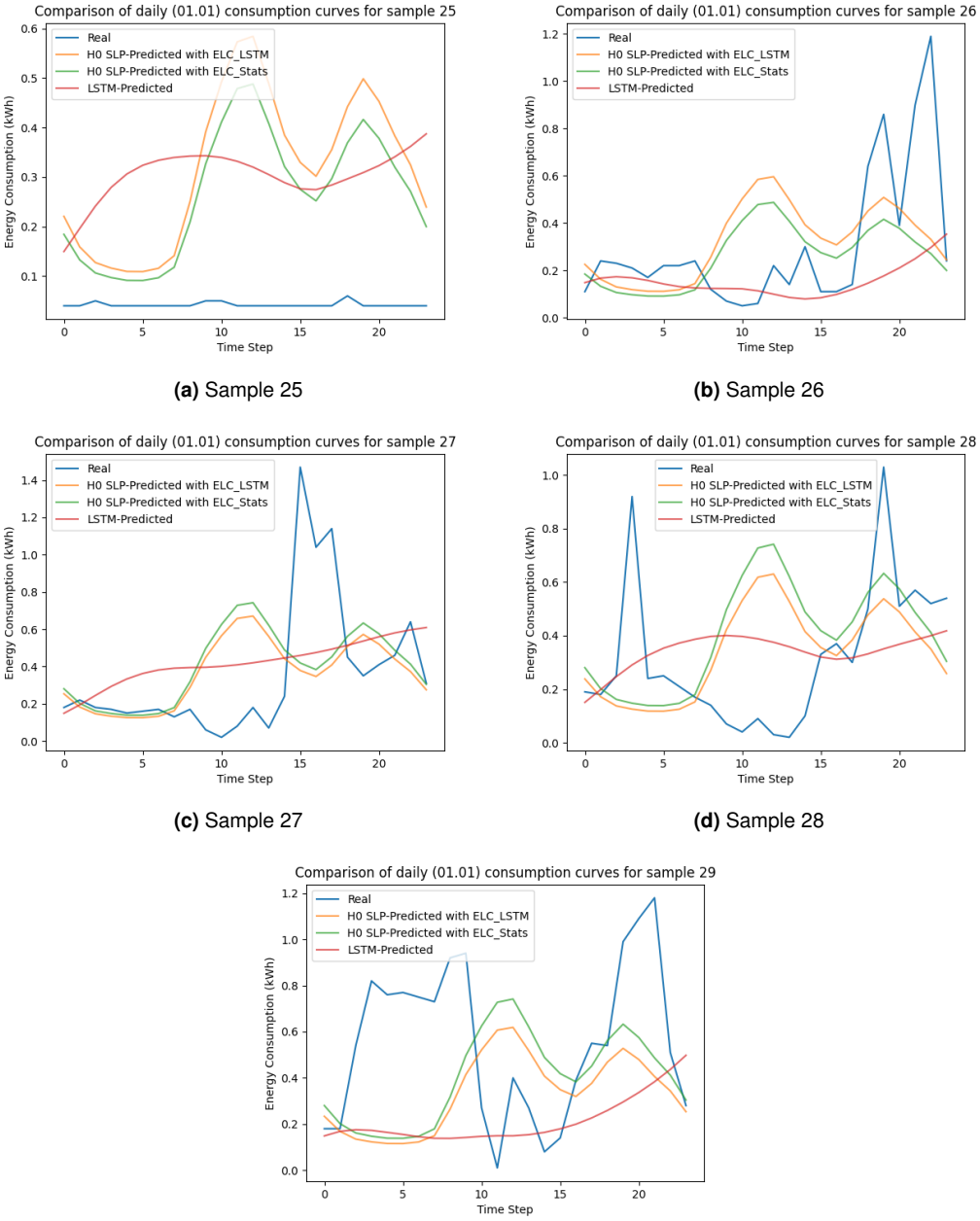


(b) H0SLP-generated curve scaled with the yearly consumption based on the number of residents



(c) LSTM-generated curve

Figure 6.3: Comparison of the electricity consumption curves for sample 27



(a) Sample 25

(b) Sample 26

(c) Sample 27

(d) Sample 28

(e) Sample 29

Figure 6.4: Comparison of the daily electricity consumption curves for 01.01.2024

Chapter 7

Conclusion

An accurate forecast of the residential electricity consumption is essential since it accounts for around 25% of the total energy consumption in Germany, and it is the basis for planning the energy system. However, predicting residential electricity consumption is challenging due to the randomness of human behavior and the lack of real data due to privacy regulations. To tackle this problem, this thesis studied integrating semantic 3D city models with previous research, statistical data, and a Deep Learning method, to simulate households, their residents, size, and electricity consumption curve, providing an alternative to simulate the power consumption at household level in Germany.

This work showed the applicability of semantic 3D city models, as they provide energy-related values such as geometrical and geographical properties. The geometrical properties combined with statistical data managed to simulate the area and the number of residents at the household level. The calculated area and the geographical location were the parameters for predicting a consumption curve by the LSTM model, which, when summed, delivered the yearly electricity consumption ELC_{LSTM}. The number of residents defined the ELC_{Stats} by relating to the statistical data on the average electricity consumption based on household size by [58]. These values were used as input for scaling the H0SLP from BDEW. Before, this load profile could only be used when the annual consumption value was known, which is normally private information. This means this thesis expanded the use of the H0SLP for simulation purposes where only the 3D City GML model is available, making the work applicable to many cases.

Deep learning techniques have shown great success in research for predicting residential electricity consumption. Therefore, an LSTM Network was developed in this thesis. The model is suitable for the task and able to work with the smart meter data. The model generated a good learning curve and an RMSE of 0.325. The results seem to capture the electricity consumption seasonality with higher values in winter and the randomness of the residential consumption. In addition, the results suggest that the model was able to relate household areas to electricity consumption. Nevertheless, the model has limitations, such as poor performance for households with an uncommon high consumption, such as in samples 25 and 26. Also, the LSTM Network predicts negative values probably due to the noisy data set. In addition, the data set is too small and unrepresentative to train a neural network. Overall, the model serves as a good basis for further research, where trying better data sets and further analyzing the network is encouraged.

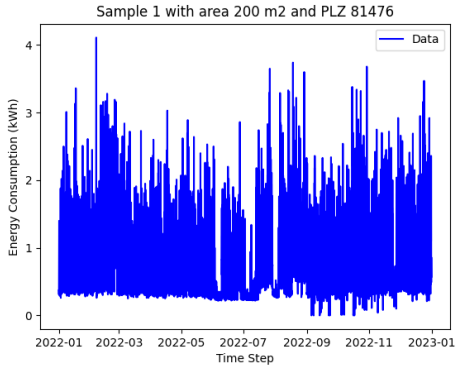
As mentioned in many studies, creating a new and more accurate official Standard load profile for households is still needed. To do this, Deep learning methods should be considered an option as they show great potential due to their adaptability and capability to capture

temporal and spatial features. In addition, the official development of a new load profile calculation method should dispose of a much bigger computational and research power than this Thesis, enabling work with consumption data from the entire country, most likely resulting in a good-performing data-driven model.

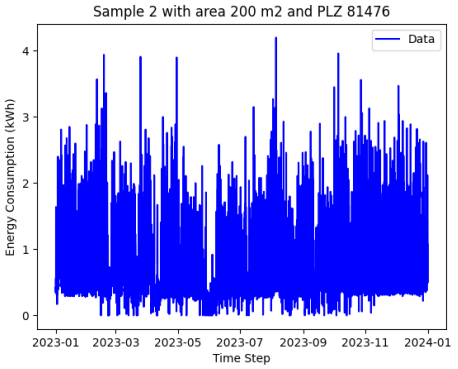
There is still room for improvement in the prediction of electricity consumption. Nevertheless, this thesis delivered a good alternative to simulate electricity consumption at the household level, with 3D city models and statistical and open data as the starting point. This work is a good basis for further research since the developed code integrates the automatization of modeling households in CityGMl with the prediction of consumption curves and their visualization. The work and the visualization support users with electricity consumption-related questions, such as sellers and buyers of electricity, grid, power plants, and storage operators.

Appendix A

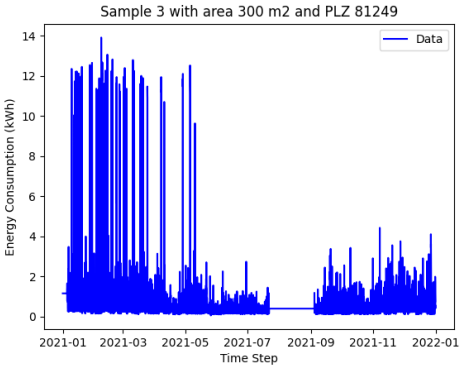
Appendix 1



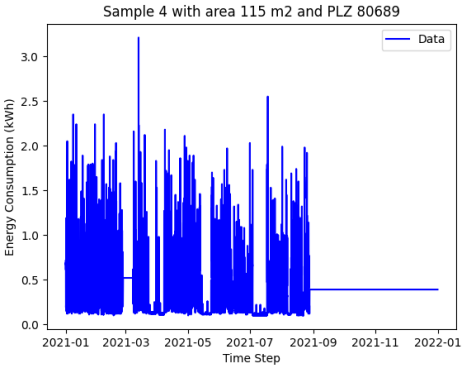
(a) Plot of Sample 1



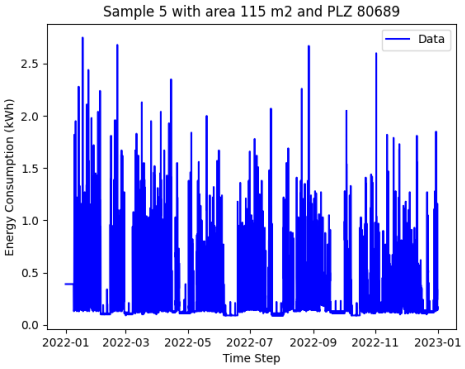
(b) Plot of Sample 2



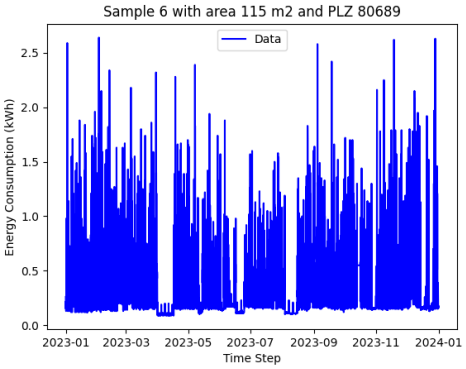
(a) Plot of Sample 3



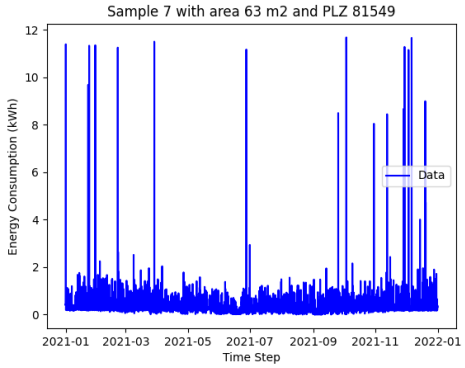
(b) Plot of Sample 4



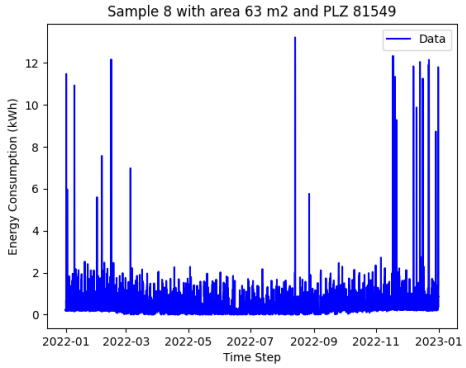
(a) Plot of Sample 5



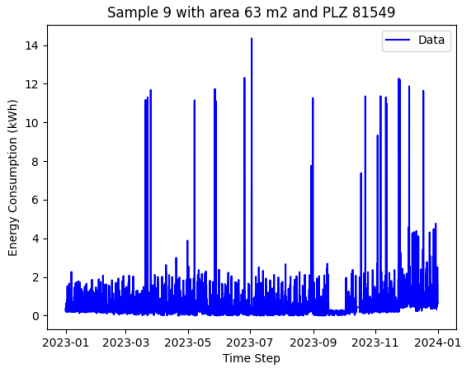
(b) Plot of Sample 6



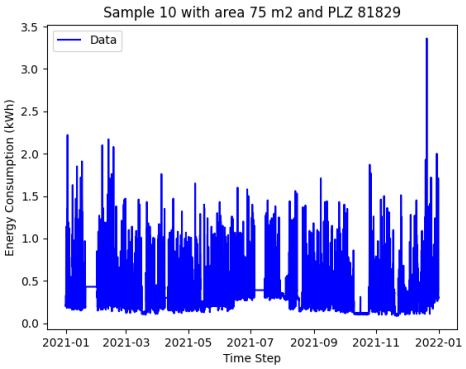
(a) Plot of Sample 7



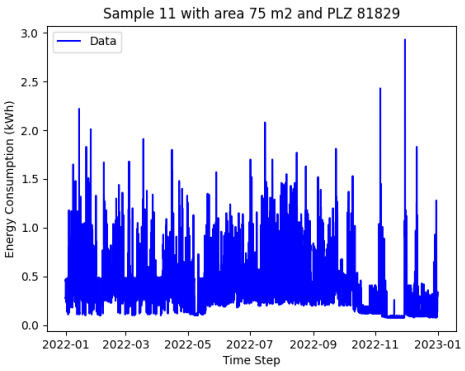
(b) Plot of Sample 8



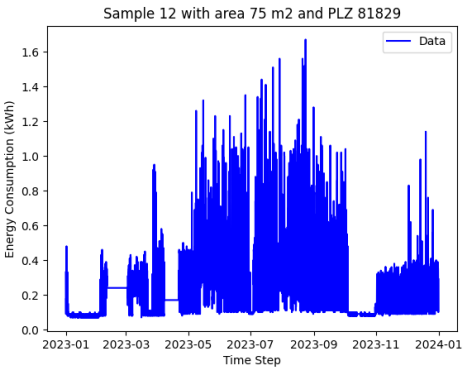
(a) Plot of Sample 9



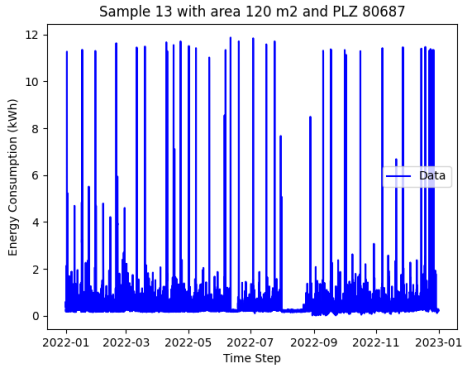
(b) Plot of Sample 10



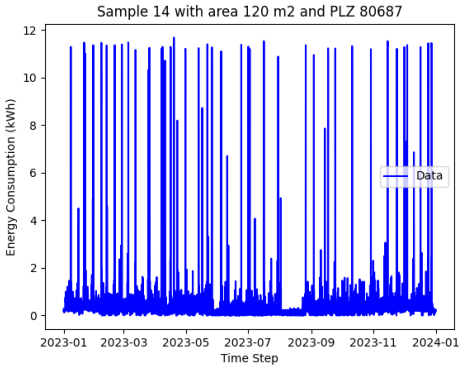
(a) Plot of Sample 11



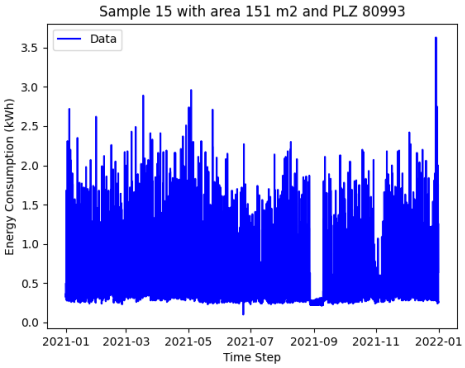
(b) Plot of Sample 12



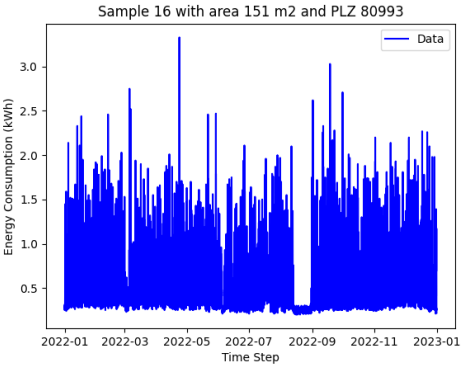
(a) Plot of Sample 13



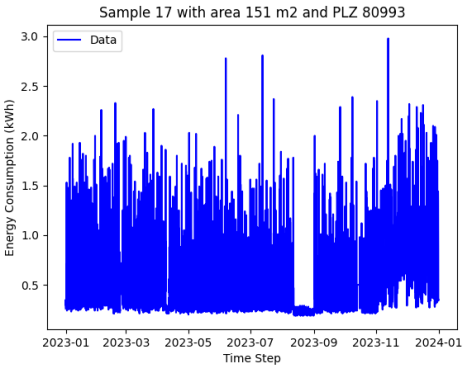
(b) Plot of Sample 14



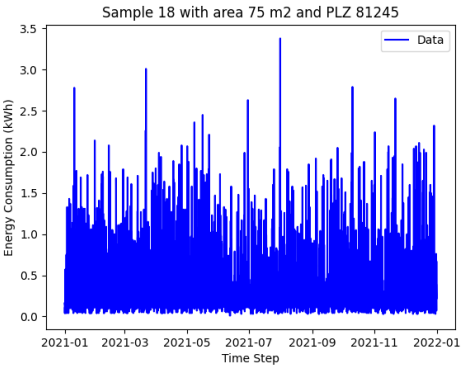
(a) Plot of Sample 15



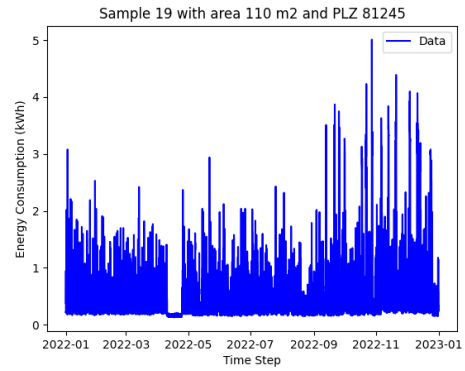
(b) Plot of Sample 16



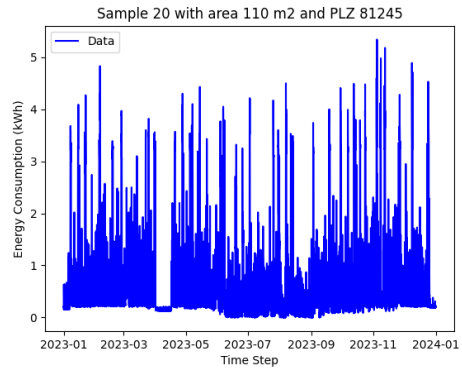
(a) Plot of Sample 17



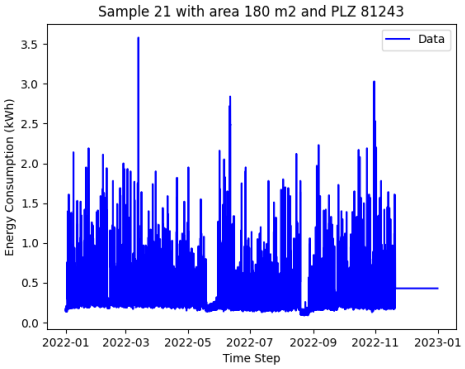
(b) Plot of Sample 18



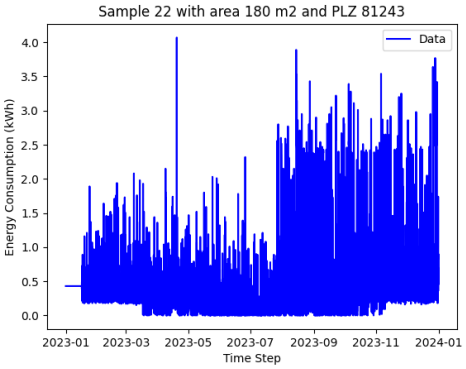
(a) Plot of Sample 19



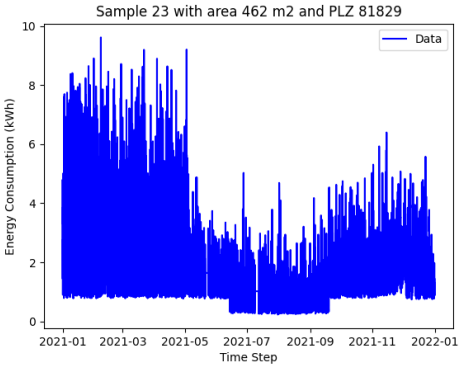
(b) Plot of Sample 20



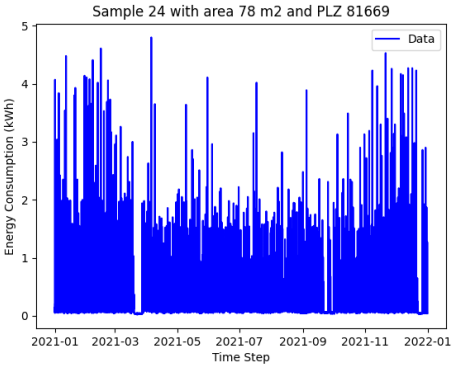
(a) Plot of Sample 21



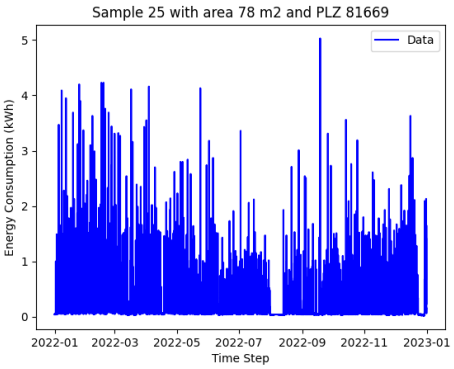
(b) Plot of Sample 22



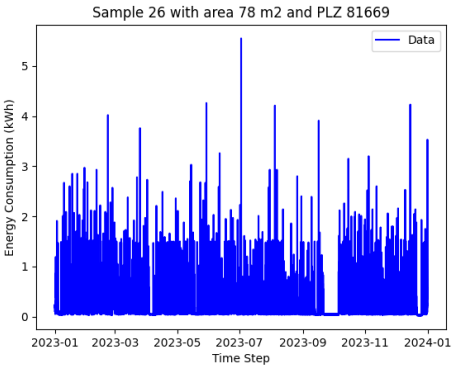
(a) Plot of Sample 23



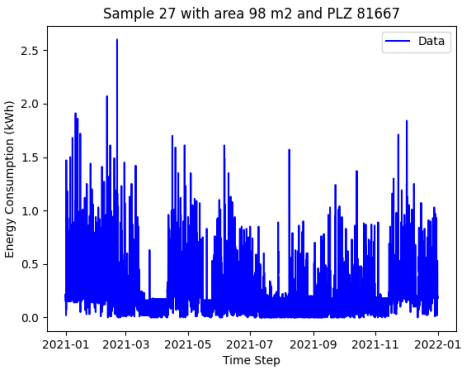
(b) Plot of Sample 24



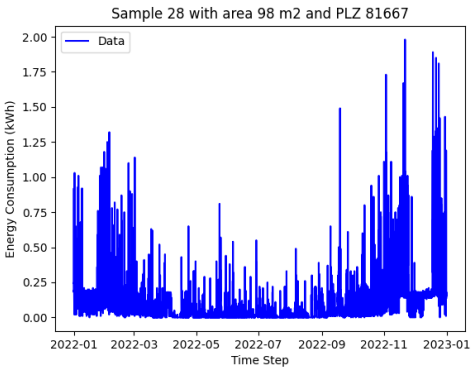
(a) Plot of Sample 25



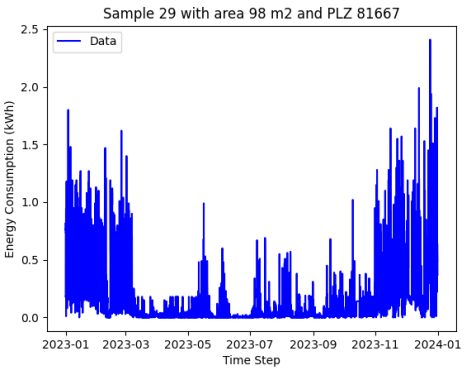
(b) Plot of Sample 26



(a) Plot of Sample 27



(b) Plot of Sample 28



(a) Plot of Sample 29

List of Figures

List of Tables

Bibliography

[1] Eurostat. *Energy Consumption in Households*. Accessed: 2024-12-05. 2023. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_consumption_in_households.

[2] Statistisches Bundesamt (Destatis). *Stromerzeugung 2023: 56 % aus erneuerbaren Energieträgern*. Accessed: 2024-12-05. Mar. 2024. URL: https://www.destatis.de/DE/Presse/Pressemitteilungen/2024/03/PD24_087_43312.html.

[3] Proedrou, E. "A Comprehensive Review of Residential Electricity Load Profile Models". In: *IEEE Access* 9 (2021), pp. 12114–12133. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3050074. URL: https://ieeexplore.ieee.org/document/9316723/ (visited on 04/24/2024).

[4] Justiz (BMJ), B. der. *Gesetz über die Elektrizitäts- und Gasversorgung (Energiewirtschaftsgesetz - EnWG)*. Accessed: 2024-12-05. 2024. URL: https://www.gesetze-im-internet.de/enwg/.

[5] Praktiknjo, A. "Sicherheit der Elektrizitätsversorgung". In: *Das Spannungsfeld von Wirtschaftlichkeit und Umweltverträglichkeit*. Available online via Springer Nature Link. Springer Nature, 2013. URL: https://link.springer.com/book/10.1007/978-3-658-04344-5.

[6] Burger, B. *Electricity Generation in Germany in 2023*. Tech. rep. Version 1, January 2024. Data corrected using Federal Statistical Office (Destatis) data. Fraunhofer ISE, Jan. 2024. URL: https://www.energy-charts.info.

[7] Shenoy, S. and Gorinevsky, D. "Risk Adjusted Forecasting of Electric Power Load". In: *Proceedings of the American Control Conference (ACC)*. Supported by a Seed Grant from TomKat Center for Sustainable Energy at Stanford University. Portland, Oregon, USA: AACC, June 2014, pp. 914–919. DOI: 10.1109/ACC.2014.6859139.

[8] Köhler, S., Betz, M., Bao, K., Weiler, V., and Schröter, B. "Determination of household area and number of occupants for residential buildings based on census data and 3D CityGML building models for entire municipalities in Germany". In: 2021 Building Simulation Conference. Sept. 1, 2021. DOI: 10.26868/25222708.2021.30573. URL: https://publications.ibpsa.org/conference/paper/?id=bs2021_30573 (visited on 07/22/2024).

[9] Behm, C., Nolting, L., and Praktiknjo, A. "How to model European electricity load profiles using artificial neural networks". In: *Applied Energy* 277 (Nov. 2020), p. 115564. ISSN: 03062619. DOI: 10.1016/j.apenergy.2020.115564. URL: https://linkinghub.elsevier.com/retrieve/pii/S030626192031076X (visited on 04/25/2024).

[10] Seim, S. "Development and application of electricity load profiles for long-term forecasting and flexibility assessment". In: (2022). URL: https://depositonce.tu-berlin.de/handle/11303/17209 (visited on 04/25/2024).

[11] Kaden, R. and Kolbe, T. H. “CITY-WIDE TOTAL ENERGY DEMAND ESTIMATION OF BUILDINGS USING SEMANTIC 3D CITY MODELS AND STATISTICAL DATA”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* II-2-W1 (Sept. 13, 2013). Conference Name: WG II/2 </br> ISPRS 8th 3D GeoInfo Conference & WG II/2 Workshop (Volume II-2/W1) - 27–29 November 2013, Istanbul, Turkey Publisher: Copernicus GmbH, pp. 163–171. ISSN: 2194-9042. DOI: 10.5194/isprsannals-II-2-W1-163-2013. URL: https://isprs-annals.copernicus.org/articles/II-2-W1/163/2013/isprsannals-II-2-W1-163-2013.html (visited on 04/22/2024).

[12] Meier, H., Fünfgeld, C., Adam, T., and Schieferdecker, B. *Repräsentative VDEW-Lastprofile - Aktionsplan WETTBEWERB M-32/99*. Tech. rep. Prepared at the Brandenburgische Technische Universität Cottbus, Lehrstuhl Energiewirtschaft. VDEW Frankfurt (Main), 1999. URL: http://www.tu-cottbus.de.

[13] Anvari, M., Proedrou, E., Schaefer, B., Beck, C., Kantz, H., and Timme, M. *Data-Driven Load Profiles and the Dynamics of Residential Electric PowerConsumption*. Sept. 19, 2020. arXiv: 2009.09287[physics]. URL: http://arxiv.org/abs/2009.09287 (visited on 04/30/2024).

[14] Hinterstocker, M. “BEWERTUNG DER AKTUELLEN STANDARDLASTPROFILE ÖSTERREICHS UND ANALYSE ZUKÜNFTIGER ANPASSUNGSMÖGLICHKEITEN IM STROMMARKT”. In: ().

[15] Krüger, A. and Kolbe, T. H. “BUILDING ANALYSIS FOR URBAN ENERGY PLANNING USING KEY INDICATORS ON VIRTUAL 3D CITY MODELS – THE ENERGY ATLAS OF BERLIN”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXIX-B2 (July 27, 2012), pp. 145–150. ISSN: 2194-9034. DOI: 10.5194/isprsarchives-XXXIX-B2-145-2012. URL: https://isprs-archives.copernicus.org/articles/XXXIX-B2/145/2012/ (visited on 04/22/2024).

[16] Silva, I. N. da, Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., and Reis Alves, S. F. dos. *Artificial Neural Networks: A Practical Course*. Cham, Switzerland: Springer International Publishing, 2017. ISBN: 978-3-319-43162-8. DOI: 10.1007/978-3-319-43162-8. URL: https://link.springer.com/book/10.1007/978-3-319-43162-8.

[17] Somu, N., M R, G. R., and Ramamritham, K. “A hybrid model for building energy consumption forecasting using long short term memory networks”. In: *Applied Energy* 261 (Mar. 2020), p. 114131. ISSN: 03062619. DOI: 10.1016/j.apenergy.2019.114131. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306261919318185 (visited on 09/11/2024).

[18] Portele, C. *OGC® Geography Markup Language (GML) — Extended schemas and encoding rules*. Version 3.3.0. OGC document 10-129r1. Open Geospatial Consortium. Feb. 2012. URL: http://www.opengis.net/spec/GML/3.3.

[19] Open Geospatial Consortium. *OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard*. Tech. rep. Available online: https://docs.ogc.org/is/20-010/20-010.html. Open Geospatial Consortium (OGC), 2021. URL: https://docs.ogc.org/is/20-010/20-010.html.

[20] Kolbe, T. H. “Representing and Exchanging 3D City Models with CityGML”. In: *Proceedings of the 3rd International Workshop on 3D Geo-Information*. Ed. by Lee, J. and Zlatanova, S. Lecture Notes in Geoinformation and Cartography. Berlin, Heidelberg: Springer Verlag, 2009, pp. 15–31. URL: http://www.springer.com/978-3-540-87394-5.

[21] Kutzner, T., Smyth, C. S., Nagel, C., Coors, V., Vinasco-Alvarez, D., Ishimaru, N., Yao, Z., Heazel, C., and Kolbe, T. H., eds. *OGC City Geography Markup Language (CityGML) Part 2: GML Encoding Standard*. Version 3.0. OGC document 21-006r2, Approved Standard. Submission Date: 2023-01-25, Approval Date: 2023-05-10, Publication Date: 2023-06-20. Open Geospatial Consortium (OGC). June 2023. URL: http://www.opengis.net/doc/IS/CityGML-2/3.0.

[22] Karlsruhe Institute of Technology (KIT), Institute for Applied Computer Science (Campus North). *CityGML Example FZK-Haus*. Accessed: 15.01.2025. URL: https://www.citygmlwiki.org/index.php?title=FZK_Haus.

[23] Biljecki, F., Kumar, K., and Nagel, C. "CityGML Application Domain Extension (ADE): overview of developments". In: *Open Geospatial Data, Software and Standards* 3 (2018), p. 13. DOI: 10.1186/s40965-018-0055-6. URL: https://doi.org/10.1186/s40965-018-0055-6.

[24] LeCun, Y., Bengio, Y., and Hinton, G. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539. URL: https://doi.org/10.1038/nature14539.

[25] Rasamoelina, A. D., Adjailia, F., and Sinčák, P. "A Review of Activation Function for Artificial Neural Network". In: *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 2020, pp. 281–286. DOI: 10.1109/SAMI48414.2020.9108717.

[26] Brownlee, J. *How to Avoid Overfitting in Deep Learning Neural Networks*. Accessed: 2024-12-05. 2019. URL: https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/.

[27] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[28] Wythoff, B. J. "Backpropagation neural networks: A tutorial". In: *Chemometrics and Intelligent Laboratory Systems* 18.2 (1993), pp. 115–155. ISSN: 0169-7439. DOI: https://doi.org/10.1016/0169-7439(93)80052-J. URL: https://www.sciencedirect.com/science/article/pii/016974399380052J.

[29] Johannesen, N. J. "Machine Learning Applications for Load Predictions in Electrical Energy Network". PhD thesis. University of Agder, 2022.

[30] Yang, L. and Shami, A. "On hyperparameter optimization of machine learning algorithms: Theory and practice". In: *Neurocomputing* 415 (2020), pp. 295–316. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2020.07.061. URL: https://www.sciencedirect.com/science/article/pii/S0925231220311693.

[31] Probst, P., Boulesteix, A.-L., and Bischl, B. "Tunability: Importance of Hyperparameters of Machine Learning Algorithms". In: *Journal of Machine Learning Research* 20 (Mar. 2019). Submitted 7/18; Revised 2/19; Published 3/19, pp. 1–32. URL: http://www.jmlr.org/papers/volume20/18-444/18-444.pdf.

[32] Brownlee, J. *Understand the Impact of Learning Rate on Neural Network Performance*. Published on September 12, 2020, in Deep Learning Performance. Accessed: 2024-12-05. Sept. 2020. URL: https://machinelearningmastery.com/understand-the-impact-of-learning-rate-on-neural-network-performance/.

[33] rendyk. *Tuning the Hyperparameters and Layers of Neural Network Deep Learning*. Last updated: November 25, 2024. Nov. 2024. URL: https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/.

[34] Kim, T.-Y. and Cho, S.-B. “Predicting residential energy consumption using CNN-LSTM neural networks”. In: *Energy* 182 (2019), pp. 72–81. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2019.05.230. URL: https://www.sciencedirect.com/science/article/pii/S0360544219311223.

[35] Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., and Zhang, Y. “Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network”. In: *IEEE Transactions on Smart Grid* 10.1 (Jan. 2019). Conference Name: IEEE Transactions on Smart Grid, pp. 841–851. ISSN: 1949-3061. DOI: 10.1109/TSG.2017.2753802. URL: https://ieeexplore.ieee.org/document/8039509/ (visited on 09/08/2024).

[36] Calin, O. *Deep Learning Architectures: A Mathematical Approach*. Springer Series in the Data Sciences. Cham, Switzerland: Springer Nature Switzerland AG, 2020. ISBN: 978-3-030-36720-6. DOI: 10.1007/978-3-030-36721-3. URL: https://doi.org/10.1007/978-3-030-36721-3.

[37] Brownlee, J. *A Gentle Introduction to Exploding Gradients in Neural Networks*. Accessed: 2025-01-10. Aug. 2019. URL: https://machinelearningmastery.com/exploding-gradients-in-neural-networks/.

[38] Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. *Dive into Deep Learning*. Available online: https://d2l.ai/index.html. 2020. URL: https://d2l.ai/index.html.

[39] *Anwendung repräsentativer Lastprofile – Step-by-step*. Accessed: 2024-12-05. BDEW (German Association of Energy and Water Industries). 2000. URL: https://www.bdew.de/media/documents/2000131_Anwendung-repraesentativen_Lastprofile-Step-by-step.pdf.

[40] Gotzens, F., Gillessen, B., Burges, S., Hennings, W., Müller-Kirchenbauer, J., Seim, S., Verwiebe, P., Tobias, S., Jetter, F., and Limmer, T. “DemandRegio - Harmonisierung und Entwicklung von Verfahren zur regionalen und zeitlichen Auflösung von Energienachfragen : Abschlussbericht”. In: (2020). Publisher: Forschungsstelle für Energiewirtschaft e. V. DOI: 10.34805/FFE-119-20. URL: https://openaccess.ffe.de/10.34805/ffe-119-20 (visited on 05/27/2024).

[41] Kaden, R. “Berechnung der Energiebedarfe von Wohngebäuden und Modellierung energiebezogener Kennwerte auf der Basis semantischer 3D-Stadtmodelle”. In: ().

[42] Köhler, S., Betz, M., and Eicker, U. “Stochastic Generation of Household Electricity Load Profiles in 15-minute Resolution on Building Level for Whole City Quarters”. In: *Energy Challenges for the Next Decade, 16th IAEE European Conference*. International Association for Energy Economics. Ljubljana, Slovenia, Aug. 25–28, 2019.

[43] Hyeon, J., Lee, H., Ko, B., and Choi, H.-J. “Deep learning-based household electric energy consumption forecasting”. In: *The Journal of Engineering* 2020.13 (2020), pp. 639–642. DOI: https://doi.org/10.1049/joe.2019.1219. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/joe.2019.1219. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/joe.2019.1219.

[44] Cascone, L., Sadiq, S., Ullah, S., Mirjalili, S., Siddiqui, H. U. R., and Umer, M. “Predicting Household Electric Power Consumption Using Multi-step Time Series with Convolutional LSTM”. In: *Big Data Research* 31 (2023), p. 100360. ISSN: 2214-5796. DOI: https://doi.org/10.1016/j.bdr.2022.100360. URL: https://www.sciencedirect.com/science/article/pii/S2214579622000545.

[45] OpenMeter. *Die Open-Data / Open-Analytics-Plattformfür Energiemessdaten und Energieeffizienz*. Accessed: 2025-01-10. 2025. URL: https://www.openmeter.de/.

[46] Barbosa, J. E. *ELC_LSTM*. https://github.com/juebarbosa/ELC_LSTM. Accessed: 2025-01-29. 2025.

[47] Landeshauptstadt München, S. A. der and Rzeha, D. P. "Bevölkerung: Privathaushalte in München 2020 - Häufigkeiten, Strukturmerkmale, räumliche Charakteristika und zeitliche Entwicklungen". In: *Münchner Statistik* 2. Quartalsheft (2021). Text, Tabellen und Grafiken.

[48] PostGIS Development Team. *Chapter 8. SFCGAL Functions Reference*. Accessed: 2024-12-07. 2024. URL: https://postgis.net/docs/reference_sfcgal.html.

[49] Achelis, D. J. and Bauministerkonferenz, F. B. der. *Ermittlung der Gebäudenutzfläche AN in der EnEV-Praxis*. Auslegungsfragen zur Energieeinsparverordnung – Teil 19, DIBt. Aug. 2014. URL: https://postgis.net/docs/reference_sfcgal.html.

[50] Loga, T., Stein, B., Diefenbach, N., and Born, R. *Deutsche Wohngebäudetypologie Beispielhafte Maßnahmen zur Verbesserung der Energieeffizienz von typischen Wohngebäuden*. 2015. URL: https://www.episcope.eu/downloads/public/docs/brochure/DE_TABULA_TypologyBrochure_IWU.pdf.

[51] (Destatis), S. B. *Statistisches Jahrbuch: Deutschland und Internationales*. Accessed: 2024-12-08. 2018. URL: https://www.destatis.de/DE/Themen/Querschnitt/Jahrbuch/statistisches-jahrbuch-2018-dl.pdf?__blob=publicationFile.

[52] Edmonds, J. "Matroids and the Greedy Algorithm". In: *Mathematical Programming* 1 (1971), pp. 127–136. DOI: 10.1007/BF01584082. URL: https://doi.org/10.1007/BF01584082.

[53] Deutschland, S. B. *Zensus München, Landeshauptstadt: Gebäude und Wohnungen sowie Wohnverhältnisse der Haushalte*. 2011. URL: https://www.destatis.de/DE/Themen/Querschnitt/Zensus/zensus-2011.html.

[54] Brownlee, J. *Long Short-Term Memory Networks With Python*. 2017.

[55] Ramachandran, P., Zoph, B., and Le, Q. V. *Searching for Activation Functions*. 2017. arXiv: 1710.05941 [cs.NE]. URL: https://arxiv.org/abs/1710.05941.

[56] Developers, T. *TensorFlow Keras API*. Accessed: 2024-12-16. 2024. URL: https://www.tensorflow.org/api_docs/python/tf/keras (visited on 12/16/2024).

[57] He, K., Zhang, X., Ren, S., and Sun, J. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

[58] Energie- und Wasserwirtschaft, B. der. *Strombedarf der Haushalte: Durchschnittlicher Stromverbrauch (ohne Heizstrom) je Haushalt nach Haushaltsgrößen*. Accessed: 2025-01-15. 2021. URL: https://www.bdew.de/media/documents/Grafik_BDEW_Stromverbrauch_Haushalte_nach_Gr%C3%B6%C3%9Fe.pdf.

[59] Database, 3. C. *3dcitydb-web-map: Cesium-based 3D Viewer and JavaScript API for the 3D City Database*. Accessed: 2025-01-10. URL: https://github.com/3dcitydb/3dcitydb-web-map.

[60] *3D City Database User Manual*.

[61] Li, P., Rao, X., Blase, J., Zhang, Y., Chu, X., and Zhang, C. *CleanML: A Study for Evaluating the Impact of Data Cleaning on ML Classification Tasks*. 2021. arXiv: 1904.09483 [cs.DB]. URL: https://arxiv.org/abs/1904.09483.

[62] Mokbel, M. F. and Aref, W. G. “Space-Filling Curves”. In: *Encyclopedia of GIS*. Ed. by Shekhar, S. and Xiong, H. Boston, MA: Springer US, 2008, pp. 1068–1072. ISBN: 978-0-387-35973-1. DOI: 10.1007/978-0-387-35973-1_1233. URL: https://doi.org/10.1007/978-0-387-35973-1_1233.

Disclaimer

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

München, January 30, 2025

(Signature)