Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Π.Μ.Σ. Πληροφοριακά Συστήματα & Υπηρεσίες

Πρόγραμμα Μεταπτυχιακών Σπουδών

Big Data & Analytics

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	Σύνοψη έξυπνου δικτύου και βραχυπρόθεσμη πρόβλεψη κατανάλωσης ηλεκτρικής ενέργειας σε οικιακούς χρήστες. An overview on smart grid and short-term residential load forecasting.
Ονοματεπώνυμο Φοιτητή	Γεώργιος Καρμοίρης
Πατρώνυμο	Παναγιώτης
Αριθμός Μητρώου	ME2015
Επιβλέπων	Μιχαήλ Φιλιππάκης

Abstract

This research has been conducted as part of the postgraduate program at the University of Piraeus, Department of Digital Systems, Big Data & Analytics. The work is divided in two main sections: overview of smart grids and short-term load forecasting for residential users. Energy grid is on a transition phase from a conventional grid to a smart one. This work presents the main differences between those two and highlights the main components and the benefits of a smart grid. The deployment of such an infrastructure and more specifically the installation of smart meter, has increased massively the amount of collected data. Providers started diving into those data in order to identify patterns and improve their services and profit margins. Artificial intelligence and machine learning technics are currently used in large scale for load forecasting, which is necessary for planning, demand response, supply-demand equilibrium. The past five years there is a high interest in load forecasting for residential consumers; a task that is very challenging due to the volatility of such data. There many different consumption patterns depending on the area, type of house, demographics of the residents, weather, existence of solar panels, existence of an electric vehicle. The data, examined in this work, was collected during the GridFlex Heeten project [99]. The data was collected between August 2018 and August 2020 in 77 households all situated in Heeten (The Netherlands) and consists of electricity consumption per minute per household. After performing an exploratory data analysis, we created individual models, SARIMAX, Vanilla-LSTM, Encoder-Decoder LSTM, for three houses and compared the results.

Keywords: Smart grid, AMI, Timeseries, short-term load forecasting, SARIMAX, LSTM, Encoder-Decoder LSTM.

Περίληψη

Η παρούσα έρευνα έχει διεξαχθεί στο πλαίσιο του μεταπτυχιακού προγράμματος στο Πανεπιστήμιο Πειραιώς, Τμήμα Ψηφιακών Συστημάτων, κατεύθυνση Μεγάλα Δεδομένα & Αναλυτική. Η εργασία χωρίζεται σε δύο κύριες ενότητες: επισκόπηση των έξυπνων δικτύων και βραχυπρόθεσμη πρόβλεψη ηλεκτρικής κατανάλωσης για οικιακούς χρήστες. Το ενεργειακό δίκτυο βρίσκεται σε φάση μετάβασης από συμβατικό δίκτυο σε έξυπνο. Αυτή η εργασία παρουσιάζει τις κύριες διαφορές μεταξύ αυτών και υπογραμμίζει τα κύρια χαρακτηριστικά και τα οφέλη ενός έξυπνου δικτύου. Η ανάπτυξη μιας τέτοιας υποδομής και πιο συγκεκριμένα η εγκατάσταση έξυπνων μετρητών, έχει αυξήσει σε τεράστιο βαθμό τον όγκο των δεδομένων που συλλέγονται. Οι πάροχοι ηλεκτρικής ενέργειας άρχισαν να εξετάζουν αυτά τα δεδομένα προκειμένου να εντοπίσουν μοτίβα και να βελτιώσουν τις υπηρεσίες και τα περιθώρια κέρδους τους. Η τεχνητή νοημοσύνη και οι τεχνικές μηχανικής μάθησης χρησιμοποιούνται επί του παρόντος σε μεγάλη κλίμακα για την πρόβλεψη κατανάλωσης ηλεκτρικής ενέργειας, κάτι απαραίτητο για τον σχεδιασμό, την απόκριση της ζήτησης, την ισορροπία προσφοράς-ζήτησης. Τα τελευταία πέντε χρόνια υπάρχει μεγάλο ενδιαφέρον για την πρόβλεψη φορτίου για οικιακούς καταναλωτές, πράγμα πολύ δύσκολο λόγω της αστάθειας τέτοιων δεδομένων. Υπάρχουν πολλά διαφορετικά μοτίβα κατανάλωσης ανάλογα με την περιοχή, τον τύπο του σπιτιού, τα δημογραφικά στοιχεία των κατοίκων, τον καιρό, την ύπαρξη ηλιακών συλλεκτών, την ύπαρξη ηλεκτρικού οχήματος. Τα δεδομένα, που εξετάστηκαν σε αυτή την εργασία, συλλέχθηκαν κατά τη διάρκεια του έργου GridFlexHeeten [97]. Τα δεδομένα συλλέχθηκαν μεταξύ Αυγούστου 2018 και Αυγούστου 2020 για 77 νοικοκυριά που βρίσκονται όλα στο Heeten (Ολλανδία) και αποτελούνται από κατανάλωση ηλεκτρικής ενέργειας ανά λεπτό ανά νοικοκυριό. Αφού κάναμε μια επεξηγηματική ανάλυση, δημιουργήσαμε μεμονωμένα μοντέλα, SARIMAX, Vanilla-LSTM, Encoder-Decoder LSTM, για τρία σπίτια και συγκρίναμε τα αποτελέσματα.

Λέξεις Κλειδιά: Έξυπνο δίκτυο, Προηγμένη υποδομή μέτρησης AMI, Βραχυπρόθεσμη πρόβλεψη κατανάλωσης ηλεκτρικής ενέργειας για οικιακούς χρήστες, Χρονοσειρές, SARIMAX, Vanilla-LSTM, Encoder-Decoder LSTM.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Φιλιππάκη Μιχαήλ για την στήριξη και τη κατανόηση που επέδειξε για το χρόνο αποπεράτωσης της εργασίας. Θα ήθελα επίσης να ευχαριστήσω την Δρ. Πούλου Μαριλένα για τη βοήθεια στην επίβλεψη της ανάλυσης των δεδομένων, τη συμβολή της στο πειραματικό μέρος της διατριβής και τα χρήσιμα σχόλιά της στην ερευνητική ανάλυση. Ευχαριστώ όλους τους καθηγητές του μεταπτυχιακού προγράμματος «Μεγάλα Δεδομένα και Αναλυτική» για όλα τα εφόδια που μας έδωσαν. Επιπλέον ένα μεγάλο ευχαριστώ στους φίλους Ιωάννη Καραγιάννη και Σταμάτη Γεωργόπουλο για τη αμέριστη βοήθειά τους και τη μετάδοση των γνώσεων τους. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου και την σύντροφό μου για όλη τη στήριξη που μου προσέφεραν όλη αυτή την απαιτητική περίοδο.

Table of Contents

1. Introduction 1

2. Background 2

2.1. Conventional Grid 2

2.2. Smart Grid 4

2.2.1. Definition 4
2.2.2. Smart Grid vs Conventional Grid 5
2.2.3. Smart Grid technologies 6
2.2.4. Data Management on Smart Grid 7
2.2.5. Pricing mechanisms in smart grid 8
2.2.6. Electric Vehicles (EVs) 9

2.3. Smart Meters 10

2.3.1. A glance in history 10
2.3.2. First generation smart meters 10
2.3.3. Second generation smart meters 11
2.3.4. Smart meter roll-out in Europe 12

2.4. Advanced Metering Infrastructure (AMI) 13

2.4.1. Definition 13
2.4.2. AMI vs AMR 13
2.4.3. High level Overview 13
2.4.4. Detailed Overview 14
2.4.5. Security Challenges 16

2.5. Load Forecasting 17

2.5.1. Supply – demand equilibrium 17
2.5.2. Forecast Horizons 17
2.5.3. Affected Stakeholders 17
2.5.4. Factors affecting load forecasting 18

3. Literature Review 19

3.1. Theoretical Background 21

3.1.1. Time Series Analysis 21
3.1.2. SARIMAX 21
3.1.3. Box-Jenkins Method 22
3.1.4. LSTM 23

4. Methodology 27

4.1. Problem definition 27

4.2. GridFlexHeeten dataset 27

4.2.1. Energy Consumption Data 27
4.2.2. Weather data 29

4.3. Data Preprocessing 30

4.3.1. Energy Consumption Data 30
4.3.2. Weather data 32

4.4. Exploratory Data Analysis (EDA) 34

4.4.1. Data Visualization after pre-processing 34
4.4.2. Data Cleaning 36
4.4.3. Time series analysis 40
4.4.4. Stationary analysis 44
4.4.5. Autocorrelation and partial autocorrelation analysis 45

4.5. Evaluation Metrics 46
4.6. Implementation 47
4.6.1. SARIMAX 47
4.6.2. Vanilla LSTM 47
4.6.3. Encoder-Decoder LSTM 48
5. Results 49
5.1. SARIMAX 51
5.2. Vanilla LSTM 54
5.3. Encoder-Decoder LSTM 57
5.4. Apply model on unseen houses 60
6. Conclusion & Future Work 62
References 64

List of Figures

Figure 1: Components of conventional grid [1]. 2
Figure 2: Conventional Grid versus Smart Grid [16] 5
Figure 3: Smart Grid ecosystem 6
Figure 4: Data management flow in smart grid. 7
Figure 5: Smart meter architecture. 10
Figure 6: Smart metering reading high level architecture. 10
Figure 7: Smart meter evolution [35]. 11
Figure 8: Advanced Metering Infrastructure (AMI). 14
Figure 9: AMI structure & Integrated technologies. 15
Figure 10: Box-Jenkins method. 22
Figure 11: LSTM block structure 24
Figure 12: LSTM walkthrough step 1. 24
Figure 13: LSTM walkthrough step 2. 25
Figure 14: LSTM walkthrough step 3. 25
Figure 15: LSTM walkthrough step 4. 26
Figure 16: Energy flow in a household. 29
Figure 17: Raw data - sample of top 200k rows. 30
Figure 18: Energy Data - Preprocessing flow 31
Figure 19: House 1 data, first 5 rows (top), column info (middle), stats (bottom). 32
Figure 20: Top 5 lines of weather data set (top), column types (bottom right), stats (bottom left) 33
Figure 21: Weather data cleaned and formatted. 33
Figure 22: House 2 (top), House 3 (middle), House 69 (bottom). Data after preprocessing. 35
Figure 23: Weather data. 36
Figure 24: Example of houses with erratic data. House 11 (top), House 17 (bottom) 37
Figure 25: House 3 data, first 5 rows (top), column info (middle), stats (bottom). 40
Figure 26: House 3 IMPORT_KW distribution. 41
Figure 27: Pearson correlation heatmap. 41
Figure 28: House 3 IMPORT_KW data 42
Figure 29: House 3 timeseries decomposition. 42
Figure 30: Distribution of energy consumption by month. 43
Figure 31: Distribution of energy consumption by week 43
Figure 32: Distribution of energy consumption by hour. 44
Figure 33: Distribution of consumption between holidays and non-holidays. 44
Figure 34: Stationary check using rolling mean and std. 45
Figure 35: Results of ADF and KPSS tests 45
Figure 36: Auto and partial correlation for energy consumption. 46
Figure 37: Vanilla LSTM model plot 48
Figure 38: Encoder-Decoder LSTM model plot. 49
Figure 39: Vanilla LSTM on aggregated model applied on houses 2 and 3. 50
Figure 40: SARIMAX model diagnostics for house 2 (top), house 3 (middle) and house 69 (bottom) 51
Figure 41: SARIMAX on house 2 (top). RMSE per day (bottom). 52
Figure 42: SARIMAX on house 3 (top). RMSE per day (bottom). 53
Figure 43: SARIMAX on house 69 (top). RMSE per day (bottom). 54
Figure 44: Vanilla LSTM on house 2 (top). RMSE per day (bottom). 55
Figure 45: Vanilla LSTM on house 3 (top). RMSE per day (bottom). 56
Figure 46: Vanilla LSTM on house 69 (top). RMSE per day (bottom). 57
Figure 47: Encoder-Decoder LSTM on house 2 (top). RMSE per day (bottom). 58
Figure 48: Encoder-Decoder LSTM on house 3 (top). RMSE per day (bottom). 59
Figure 49: Encoder-Decoder LSTM on house 69 (top). RMSE per day (bottom). 60
Figure 50:Vanilla LSTM house 2 model applied on house 66 (top). RMSE per day (bottom). 61
Figure 51: Vanilla LSTM house 3 model applied on house 4 (top). RMSE per day (bottom). 62

List of Tables

Table 1: Comparison of 1G & 2G smart meters [35].....11
Table 2: Status of smart metering roll-out [36].....12
Table 3: Overview of missing values count.....37
Table 4: House data statistics after replacing missing data.....39
Table 5: Summing up results of short-term load forecasting.....63

List of Symbols

ACF	Autocorrelation Function
ADF	Augmented Dickey-Fuller
AIC	Akaike's Information Criterion
AMI	Advanced Metering Infrastructure
AMR	Automatic Meter Reading
AR	Autoregressive
DR	Demand Response
DSM	Demand Side Management
DSO	Distribution System Operators
EDA	Exploratory Data Analysis
EV	Electric Vehicle
HAN	Home Area Network
I	Differencing
ICT	Information and Communications technology
KPSS	Kwiatkowski–Phillips–Schmidt–Shin
LSTM	Long Short-Term Memory
LTLF	Long-Term Load Forecasting
MA	Moving Average
MAPE	Mean Absolute Percentage Error
MDMS	Meter Data Management System
MTS	Multivariate Time Series
MTLF	Medium-Term Load Forecasting
PACF	Partial Autocorrelation Function
PV	Photovoltaic
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SM	Smart Meter
STLF	Short-Term Load Forecasting
WAN	Wide Area Network

1. Introduction

Electric energy plays an utmost role in humanity past, present and future. Energy demand is growing exponentially and the grid should go through a transformation in order to scale respectively. In the past years, smart grid technology has developed vigorously. Information and Communications technologies (ICTs) are incorporated into the grid and are responsible for its digitalization. Smart meters have also been widely deployed and drastically improved the observability of power grid. They also helped in accumulating a large amount of data, which laid a solid foundation for the application of various forecasting models. The increased penetration of renewable energy sources brings many challenges to the existing energy grid; hence the transformation should be accelerated. Electric energy is difficult to store, therefore on one hand should not be supplied in excess of demand and on the other hand should not be in short. First case results in wasting of energy, second case may cause outages. Load forecasting become very important to maintain the balance between energy supply and demand; supply – demand equilibrium [1].

Forecasting is the process of making predictions, based on past and present data and by analyzing trends/patterns. Load forecasting refers to the prediction of electricity demand. It began in the 1980s, mainly done using manual calculations by field experts. As the grid was becoming more advanced and complex achieving high accuracy became more and more difficult. The rapid development of machine learning and artificial intelligence played an important role in improving the accuracy of prediction. An accurate forecast plays an essential role in the upfront planning of generation facilities, controlling distribution systems, managing efficiently transmission lines, demand response (DR) programs, participating in day-ahead electricity market [2].

Based on the lead time, load forecasting has three main horizons: short-term load forecasting (STLF), medium-term load forecasting (MTLF) and long-term load forecasting (LTLF) [3]. LTLF refers to horizon of months or even years and is necessary for maintenance, evolution, and scheduling of the energy grid. MTLF spans from one week to a year and it is crucial for fuel scheduling and utility assessments. STLF primarily spans over a few hours to a few weeks and it's important for grid's day to day operations. In this work we will focus on STLF and more specifically on residential users. The versatile and erratic nature of residential load consumption makes forecasts quite challenging. Therefore, more exogenous features need to be considered, e.g., square meters of the household, type of household, demographic information of the residents, weather.

This work is organized as follows. chapter 2 presents a comprehensive overview of smart grids. Chapter 3 presents the literature review on the field and on top educates the reader to some technical concepts like timeseries data, Box-Jenkins methos, SARIMAX models, Long Short-Term Memory (LSTM) model. Thereafter, chapter 4, named Methodology, provides the problem definition, describes the data, and proceeds with the analysis. Additionally, dives in some implementation details. Chapter 5 presents the results and comments on them. Finally, chapter 6 concludes the work and describes the intended future work.

2. Background

2.1. Conventional Grid

A conventional electrical grid is an interconnected network for electricity delivery from a small number of producers to a big number of consumers. Their size varies and could cover whole countries or potentially whole continents. These grids were designed in 1950s and implemented throughout 1960s and 1970s [1]. The main components of such a grid, shown in Figure 1, are:

- **Generation**. Electricity generation is the process of generating power from an energy source, fossil fuels, solar power, water, wind, geothermal power. The factory responsible for that is called power station and is often located near the energy source and away from heavily populated areas.
- **Substations**. Their main responsibility is to either transform voltage from high to low (step down) or from low to high (step up).
- **Transmission**. Electric power transmission is the movement of the generated energy from the source to the substations. For efficient transmission in long distances, the wires installed are capable of carrying high voltages and low amperages.
- **Distribution**. Electric power distribution to individual customers, where voltage is stepped down again to the required service voltage(s).
- **Storage**. Electrical energy is stored when demand is low and returned to the grid when demand increases.

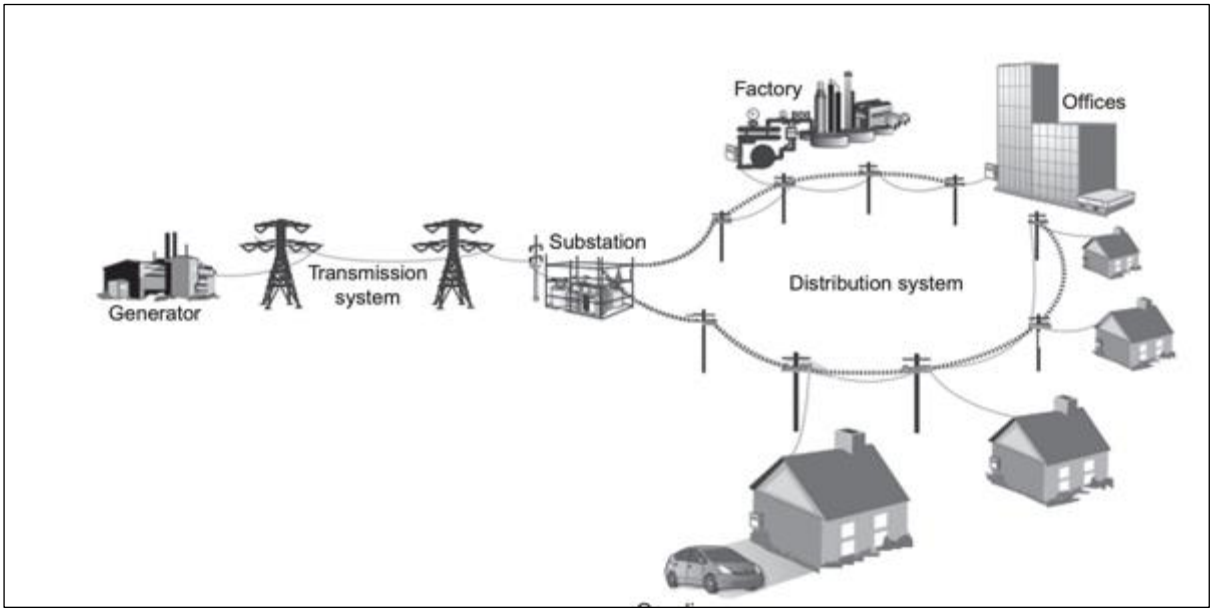

Figure 1: Components of conventional grid [1].

The infrastructure of a conventional power grid is electromechanical, mechanical devices that are electrically operated. Such a setup has though no means of communication between deployed devices and very little internal regulation. The centralized and unidirectional notion of this grid makes it difficult, if not impossible, adjusting dynamically the energy flow from generators to consumers. The consumers on the other hand have a very restricted ability to integrate their own energy storage and generator cells into the grid. Also, they have very little insights regarding their their monthly electricity usage, bills and energy prices.

In the beginning, none of this much mattered since the the electric power industry was blessed with enormous growth and scale. Making use of rate of return regulations, still in effect in many parts of the world, the major players of the industry were investing on building bigger power plants and more transmission and distribution lines. Fossil fuel power generation in combination with the growth in economies resulted in a rapid rise in demand and kept the prices low. Under these circumstances, suppliers didn't even bother with energy efficiency or promoting demand management. On the contrary consumers were encourages to use more introducing also falling block tariffs, meaning the more the consumer's use the lower per unit cost [6].

This setup was planned and installed many years ago; at times that energy generation was based on fossil fuels and consumer profiles were totally different. Hence the disadvantages of conventional grids are exposed in a faster pace. Let's state some of the disadvantages of such a grid [5]:

- Centralized power generation. This eliminates the possibility of easily integrating alternative energy (renewable) sources into the grid.
- Unidirectional network. Energy flows from generators to end consumers without the ability to get feedback from the other side. This means that the conventional power grid is not flexible in adjusting to the growing energy demands, facing challenges in locating grid failures, spontaneously rerouting electricity, and overheating of power lines.
- Installed grid's electromechanical elements were designed to meet historical energy demands rather than the current demand.
- Inefficient load balancing. Load balancing is about keeping the demand curve in sync with the generation curve. If demand exceeds supply, then the grid collapses and electric power is not available to any user. Whenever supply exceeds demand, the result is unused energy and/or waste.
- Small number of sensors. This makes it difficult to pinpoint the root cause and the location of a problem.
- Monitoring electricity flows remains largely manual.
- Manual restoration. Technicians have to physically go to the location of the failure.
- Frequent failures and blackouts: outages have become common due to natural disasters, weather, and technical issues with grid controls; these outages increase risks of harm and loss.
- Fewer customer choices. Customers don't have access to flexible energy plans, don't have insights on their consumption and cannot choose their source of energy.

To address these issues, countries have started replacing and modifying the current grid to make it a smarter and more adaptive. The smart grid offers solutions to many of the problems described above.

2.2. Smart Grid

The idea of our current electric grid was conceived more than 100 years ago, when electricity demands were limited. Homes had only small energy needs such as a few light bulbs, radio and a tv, hence power generation was built around these communities. The main concern of the utilities was to deliver energy to consumers and bill them once a month. This one-way interaction model is not able to keep the changing pace on energy demand of the 21rs century. Smart grid is considered the replacement of this aging infrastructure; it can be perceived as a network that IoT devices, communication protocols, data gathering, and data management tools are working together to build a more reliable, efficient, green, and secure grid. Smart grid introduces a two-way communication between the utility and the customer enabling efficient energy management on both sides. Smart meters, smart devices, thermostats, electric vehicles (EVs) are forming a Home Area Network (HAN) which connects to an energy management system, so devices adjust their run schedule considering peak times and availability of electricity.

A very important feature introduced by smart grids is efficient demand management. Utilities turn power plants on and off depending on the amount of power needed at certain times. Peak hours are the most challenging since more power plants need to be run to meet the higher demand, hence the cost is higher. Smart grid enables utilities to manage and moderate usage with the cooperation of consumers especially on peak hours; devices run at other times deferring electricity usage from peak times, which leads to operating cost reduction. Keeping the balance between energy production and consumption is very crucial for the grid, smart grid provides near real-time insights regarding electricity demand reducing outages and evenly distribute electricity production throughout the day. Additionally, grid engineers will be able to more precisely and predictably manage electricity production reducing the need to fire up costly secondary power plants.

Conventional distribution system routes power from the utility to residential and commercial customers through power lines, switches, and transformers, relying typically on complex power distribution schemes and manual switching to keep power flowing to customers. Any break in this system caused by storms, bad weather or sudden changes in electricity demand can lead to outages. Smart grid evolves the distribution system introducing intelligence; energy fluctuations and outages are countered by automatically identifying problems in power delivery. This distribution intelligence is key for prediction in electricity usage which leads to lower production cost.

Another important change is that smart grid integrates distributed energy resources. Renewable resources such as wind and solar are a sustainable and growing source for electric power, however renewable power sources are variable by nature and add complexity to normal grid operations. Smart grid provides the data and automation needed to enable solar panels and wind farms to put energy onto the grid and optimize its use to keep up with constantly changing energy demands.

2.2.1. Definition

What is a Smart Grid? Writing a concise definition is not as easy as it sounds; the concept is relatively new and due to different designs of existing conventional grid there are various alternatives on the components used to compose such a network [7]. There are several; public and private sector, organizations, authors, and acts that have defined/visualized a smart grid: U.S. Department of Energy (DOE) [8], Electric Power Research Institute (EPRI) [9], Energy Independence and Security Act of 2007 (EISA-2007) [10], European Union Commission Task Force for Smart Grids [11], ABB [12] etc.

National Institute of Standards and Technology (NIST) released the Smart Grid Framework, where common requirements of such networks are summarized [12]:

- *Improves the reliability of the power delivery system.*
- *Optimizes facility utilization and averts construction of backup (peak load) power plants.*
- *Enhances capacity and efficiency of existing electric power networks.*
- *Improves resilience to disruption.*
- *Enables predictive maintenance and self-healing responses to system disturbances.*
- *Facilitates expanded deployment of renewable energy sources.*
- *Accommodates distributed power generation resources.*
- *Automates maintenance and operation.*
- *Reduces greenhouse gas emissions by, for example, enabling the use of electric vehicles and new power sources.*
- *Reduces oil consumption by reducing the need for inefficient generation during peak usage periods.*

- *Presents opportunities to improve grid security.*
- *Enables new energy storage options.*
- *Increases consumer choice.*
- *Enables new products, services, and market.*

Considering the common characteristics/requirements of available definitions and/or visions, this work states the following definition:

"Smart Grid is a modernized electric system that combines bidirectional communication between all components involved; production, transmission, distribution and consumption/market, in order to ensure an adaptive, interactive, predictive, secure, optimized and scalable network."

2.2.2. Smart Grid vs Conventional Grid

The smart grid is an innovative concept that will revolutionize the transmission, distribution and conservation of energy. Figure 2 depicts the main differences between a conventional and a smart grid. The current electric power delivery system (left side) has a top to bottom (one-way) approach. Production is based at a limited number of large power plants which distribute the energy from a centralized transmission system. This is almost entirely an electromechanical system; the use of sensors is limited; the electronic communication is minimal and there is almost no electronic control. Customers remain passive and just pay their bills. To be more precise, bills are calculated based on only one piece of data a monthly or quarterly kWh consumption figure multiplied by cents [14].

On the contrary, a smart grid (right side) advances many small power producers and the distribution system is decentralized/distributed. The communication in all stages is bidirectional by making use of Information and Communications technologies (ICTs). The grid is interconnected not only with energy cables, but also with communication ones to enable data flowing. Sensor networks are employed to make such a network functional, improve transparency and increase reliability and efficiency. Due to such an infrastructure, grid operators receive near to real-time information regarding energy consumption and can optimize their distribution management providing either more or less electricity from one or another source [15]. Another big change is that consumers are not passive anymore. They have the option to participate in the energy generation e.g., rooftop photovoltaics (PVs) or even in energy storage using batteries. On top, they are informed which device consumed how much energy, resulting in better consumption behaviors.

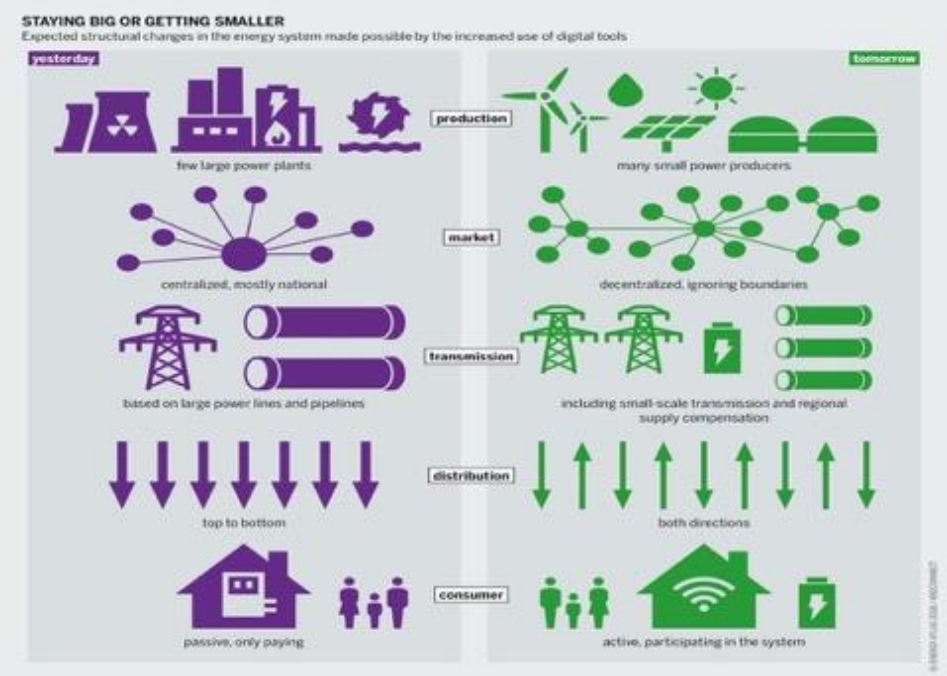

Figure 2: Conventional Grid versus Smart Grid [16]

2.2.3. Smart Grid technologies

Smart grid makes use of information technology to achieve efficiency and reliability. Such a grid consists of power generation and transmission utilities, sensing devices, smart meters and information gateways that operate in near real-time [17]. Sensing devices are responsible of monitoring performance and detecting of operational glitches, upon failure control messages are transmitted to operator's control center. The communication between devices that close to the households and the utility is routed via intermediate devices, like gateways. Gateways (also called concentrators) are responsible of collecting the data of smart meters and sensing devices and communicating this information to the utility using Wide Area Network (WAN) connection. As we have mentioned earlier there is a two-way communication between all stages meaning that utility can send control messages back to smart meters or sensing devices using the same infrastructure.

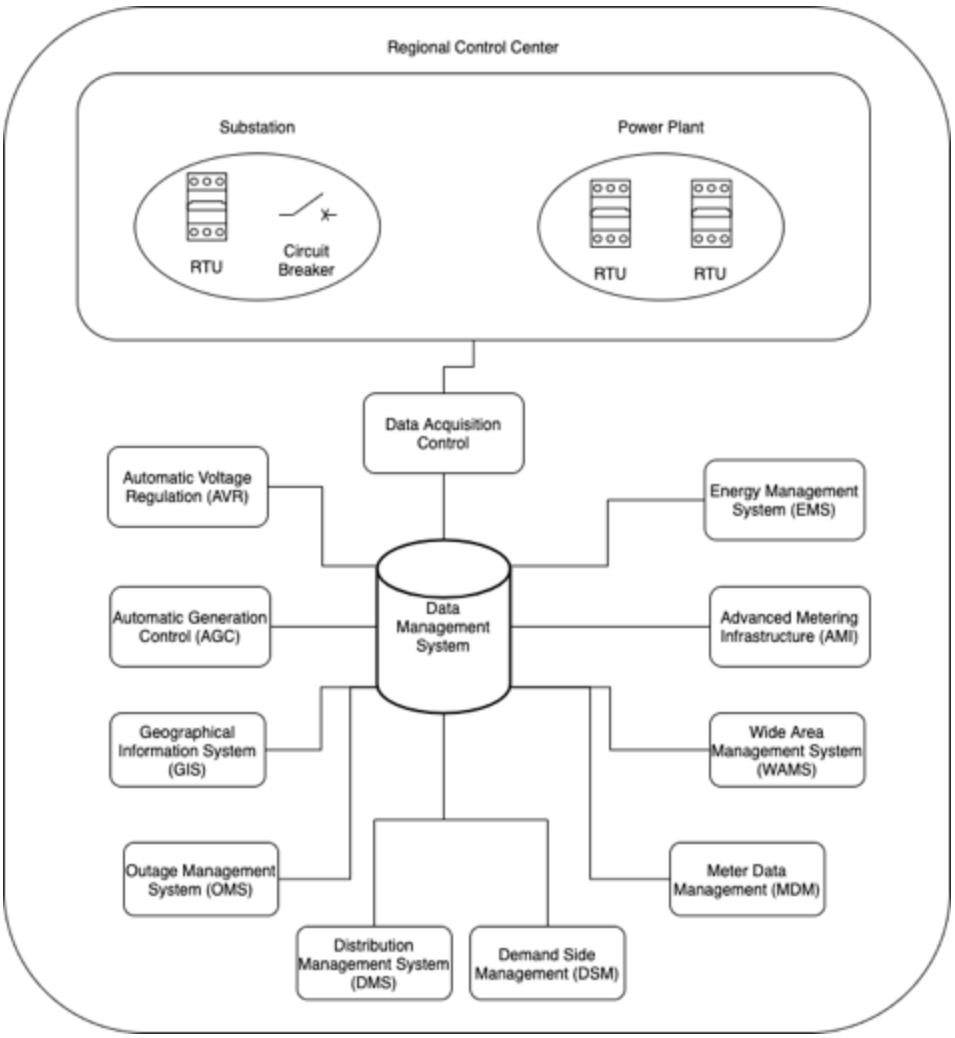

Figure 3: Smart Grid ecosystem

Figure 3 depicts smart grid's ecosystem which performs data collection and control of electricity delivery. A data acquisition control system gathers the data from substations (RTUs, circuit breaker, log servers, human machine interfaces, communication devices and gateways [18]) and power plants and outputs them into a data management system. The ecosystem incorporates several technologies such as Automatic Voltage Regulation (AVR), Automatic Generation Control (AGC), Energy Management System (EMS), Advanced Metering Infrastructure (AMI), Geographical Information System (GIS), Outage Management System (OMS), Meter Data Management (MDM), Distribution Management System (DMS), Wide Area Management System (WAMS), and Demand Side Management (DSM) [18].

AVR keeps the voltage profiles within the preconfigured limits, whereas AGC optimizes load distribution among generating units. EMS acts like an optimizer for the entire network, keeps the network reliable and secured the operating points for supervisory control and data acquisition. AMI is an integration of smart meters,

communication networks and management systems, so a two-way communication channel is opened between consumers and operators. GIS provides the infrastructure so the data can be integrated with geographical maps. OMS is responsible of automatic (if possible) restoration of the grid in case of outages. MDM is crucial on data flow and the decision-making process. DMS monitors and controls the distribution network, control messages to devices are spawned in case of needed action, hence the network is proactive. WAMS helps in grid synchronization in high voltage network and collects time measurements using Phasor Measurement Units (PMUs). DSM assists in load consumption management to improve energy efficiency and it is implemented through demand response (DR).

2.2.4. Data Management on Smart Grid

Data in a conventional grid are mainly limited to demand, voltage, and current data. On the contrary, smart grid has to deal with big data since it collects near real-time data through a huge number of data points. Examples of such data points are smart meters, sensors, weather forecasts, load profile patterns. Once the data are collected, data analysis takes place in order on one hand to help utilities efficiently manage the network and restore faulty networks and on the other hand to assist consumers to adjust their consumption (especially on peak times) to reduce cost. It is important to mention here that dealing with sensitive data brings new challenges such as data security, data privacy, data storage, data analysis, data retrieval. Figure 4 shows the data management flow in smart grids, where the data are collected, preprocessed, integrated, stored, analyzed, visualized in order to improve decision making.

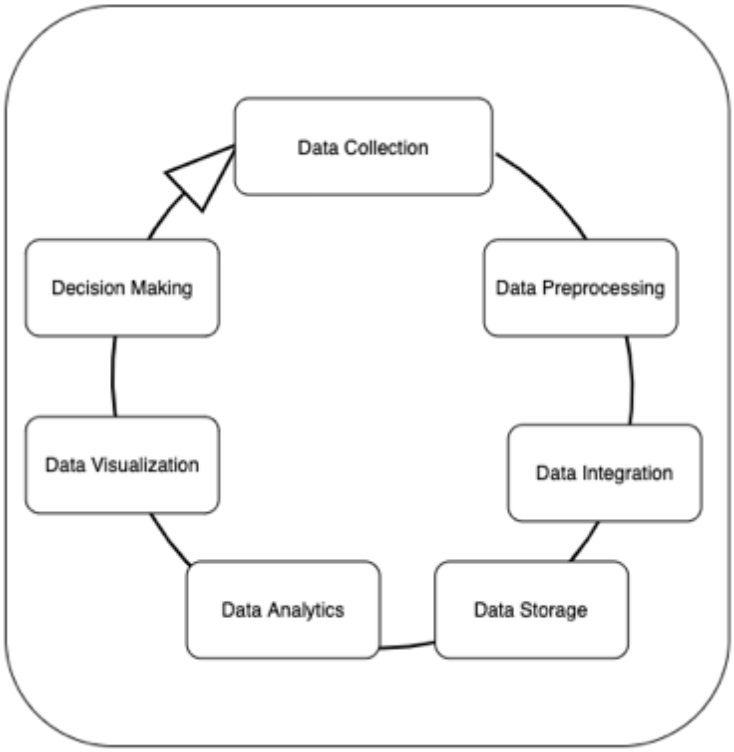

Figure 4: Data management flow in smart grid.

2.2.4.1. Data Collection

Data collection is the first step of the data management process, data are collected in frequent intervals (5 - 10 mins). The main contributor of data is the advanced metering infrastructure (AMI) where the end user data are being reported. As mentioned earlier though within the network there are many other data points; sensors, power metrics, mobile terminals, control devices, field engineers etc. [19]. Data collection process should ensure the following [20]:

- Standardization. Data should be collected in a pre-configured format.
- Reliability. Data should not be altered by any means.
- Security. The process should be secured against malicious actions.
- Privacy. Sensitive information should be exposed.
- Storage. Data should be efficiently stored for further analysis.

- ❖ Scalability. New data points, hence more data could be easily integrated into the network.

2.2.4.2. Data Preprocessing

Before performing any task of analysis, we need to check if the collected data are incomplete, inaccurate or need to be filtered, this process is known as data cleansing. There are five steps involved once erroneous data are received, data are defined, identified, corrected, documented, and modified so future faults will be avoided [21]. Another important contribution of data preprocessing is to identify and eliminate redundant or repeated data, which require more storage and add more cost.

2.2.4.3. Data Integration

Data collected from several data points are not uniform and, in many cases, need to be first integrated before analysis. For instance, consumer's daily usage data need to be integrated with weather data to help utilities perform a next day load forecasting [22].

2.2.4.4. Data Storage

Data should be stored in a database system in order to access/query them at any time. Data storage process phases three main challenges; the amount of data, the versatility of those data and the throughput (speed of processing input/output) [24]. An efficient way to address those challenges could be to use a graph storage [25].

2.2.4.5. Data Mining and Data Analytics

These processes try to uncover the hidden power of the massive amount of data gathered in a smart grid. There are operations; load forecasting, customer behavior analysis, customer profiling, that are not urgent and operations; smart meter data analysis, faults analysis, that need to be analyzed as quickly as possible. For some of these operations machine learning comes to the rescue [26].

2.2.4.6. Data Visualization

A visual representation on the results of data analysis is more straight forward to understand and leads to better decision-making [27]. In most cases, graphs reveal patterns that help pinpointing network issues or opportunities. Consumers and utilities are able to see in a visual way the end-user electricity consumption, efficiency of renewable sources, and power quality data.

2.2.4.7. Decision-Making

All steps above allow a real-time and automated decision-making process. System engineers have a real-time overview of the network and are able to isolate and fix proactively faulty sections. The system is capable in many cases to take decisions on its own and perform self-healing operations [28]. Other important features can be unlocked, such as real-time pricing, on-demand renewable generation, and capacity constraints estimation.

2.2.5. Pricing mechanisms in smart grid

Conventional price model allowed utilities to set prices that covers their operating costs plus "some" profit. Some though is vague and cannot easily controlled by any public or private entity. Hence, there are arguments that this model lacks transparency, passes utilities cost mistakes to consumers and promotes monopoly. Public utilities are often close to governments and politics might intervene for on entity or another [29]. International Energy Agency (IEA) indicated that restructuring is necessary to encourage competition. The idea is to shift from the vertically integrated structure towards an open market where consumers have the choice of power suppliers [30].

Smart grid introduces a new pricing framework called, dynamic pricing, where consumers are encouraged to participate in demand management so they can reduce their bills. The main objective is to spread the load across the day so peak hours are limited. The most popular dynamic pricing models are Time-of-Use (TOU), Real-Time Pricing (RTP), Critical Peak Pricing (CPP), and Day-Ahead Pricing (DAP)

2.2.5.1. Time-of-Use

TOU [31] is the most common pricing scheme due to its simplicity and the fact that consumers are used to fixed tariffs. TOU is a time-dependent pricing model, where different tariffs for different instants of day or season are set. Utilities apply higher rates during peaking periods, hoping that the demand will shift from peak hours to off-peak.

2.2.5.2. Real-Time Pricing

RTP is a dynamic pricing scheme that depends on the spot price of the wholesales market [32]. Once the price is finalized, a signal is sent to the retailers according to the market timeframe, day ahead, an hour ahead. The retailers notify the consumers for the prices so they can adjust their consumption accordingly. The risk/challenge of this scheme is the availability of the consumer to take the right action once the signal is received, hence an advanced infrastructure is required to automate these actions and eliminate those risks.

2.2.5.3. Critical Peak Pricing

CPP introduces a penalty for using energy within peaking periods that are known beforehand [33]. This model is less dynamic; however, it is considered an augmented TOU with the addition that critical events are announced a day ahead. The main challenge for the utilities is to set the right tariff. If prices are set high, consumers might not shift their demand. On the contrary, if prices are set low, consumers might not respond to new price signals.

2.2.5.4. Day-Ahead Pricing

DAP is a time-dependent scheme that is set day ahead. Consumers find this model attractive since they plan their energy consumption. The challenge for this pricing model is set the optimal price beforehand. Several factors [26] should be considered for this action; load forecasting, weather forecasting, supply availability, and energy price forecasting. The risk for the utilities is that they might have a loss if the peak hours occur during low price periods.

2.2.6. Electric Vehicles (EVs)

Consumer's energy profile is changing drastically; smart appliances/devices are installed in households, rooftop panels are deployed, electric vehicles are replacing conventional cars. A huge challenge for the network will be the increasing number of electric vehicles. Charging such a vehicle is energy demanding and if the network cannot handle the charging process efficiently an overload might occur [34]. It is crucial for Distribution System Operators (DSOs) to build a communication and control system to handle charging according to the grid constraints and customer's needs.

Smart charging is the mechanism of controlling time and rate at which the car is charged. This mechanism involves signals from the operator, enabling the car to stop charging (especially on peak hours) on demand. A Home Area Network (HAN) receives the control signals to balance the demand for electricity across the household and prioritize between the plugged-in vehicle or other appliances. A very interesting concept is vehicle-to-grid (V2G) [35] charging. The main idea of this technology is that the car can be turned to a temporal storage system, which supplies power to the grid at peak times and the vehicle can continue charging when the demand is reduced.

2.3. Smart Meters

A smart meter is an electronic device which records data such as electricity consumption, voltage levels, power factor and current. It reports data in short intervals (e.g., every 5 mins), allowing DSOs to analyze the data, identify consumption patterns, manage the network efficiently, automate billing [10]. The bidirectional data communication capabilities of such device allow operator to transmit control commands for better electricity management.

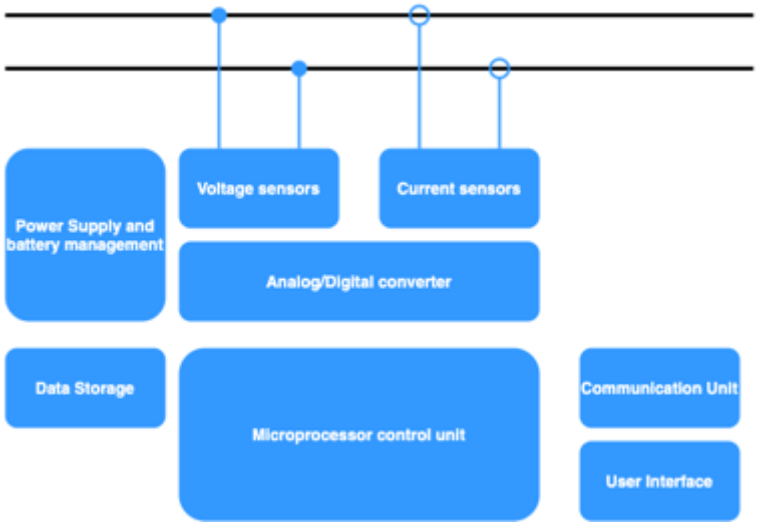

Figure 5: Smart meter architecture.

Figure 5 depicts smart meter's architecture, a power supply gives life to the meter so the meter can gather information through voltage and current sensors. An analog-digital converter passes the data to the processor control unit which coordinates data storage, the user interface, and the communication unit.

2.3.1. A glance in history

The first known meter was created in 1872 from Samuel Gardiner. The meter was counting the time energy was supplied to a set of lamps. Thomas Edison made use of the electrochemical effect of DC current to patent his first meter in 1881. A decomposing, due to the passing current, strip of copper was measured at start and at the end, with the difference being equivalent with the consumed electricity. In 1889, Otto Titusz Blathy came up with an electric meter for alternating current. Similar induction meters are still manufactured today due to their low cost and reliability. In the second half of the 20th century electronic meters and remote metering was introduced [36].

2.3.2. First generation smart meters

First generation smart meters (1G) share information between the meter and the Head-End-System (HES) [37]. Figure 6 shows the high-level steps of such communication. Initially a connection is made between a smart meter and a secondary substation with low voltage (LV)/ medium voltage (MV) transformers where data concentrators are located. These concentrators receive, process, and reassemble data from potentially thousands of smart meters and forward them to HES.

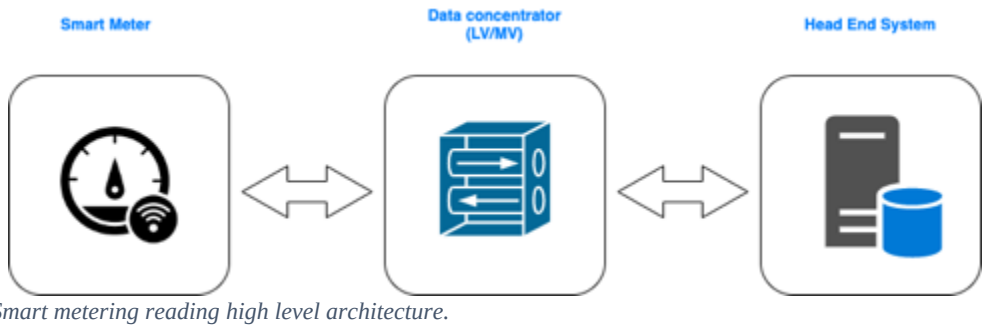

Figure 6: Smart metering reading high level architecture.

Communication in most cases is performed through Power Line Communications (PLC) or Transmission Line Communication (TLC) since these technologies can be deployed using the existing infrastructure and are reachable in installations deep in buildings. Alternatives such as Meter-Bus (WM-Bus); wireless communication, is more difficult to use since a separate network need to deployed and additional frequency planning is required.

2.3.3. Second generation smart meters

Second generation smart meters (2G) are meeting the demands of the future smart grid. Near real-time data are collected and transmitted to the DSO so they provide services such as home automation, real-time billing schemes, demand response programs, customer awareness. The frequency, the accuracy and the precision of the collected data makes the difference, Table 1, between the two generations [38].

Table 1: Comparison of 1G & 2G smart meters [35].

Metering Data	1G Smart Meter	2G Smart Meter
Active energy withdrawn	3 values per month	15 min
Active energy Injected	3 values per month	15 min
Reactive energy withdrawn	3 values per month	15 min
Reactive energy Injected	3 values per month	15 min
Active power withdrawn	30 min (peak)	15 min (peak) and instantaneous value (1s)
Active energy Injected	No	15 min (avg)
Min/max voltage	Only occasionally	1 per week
Voltage in limits	Only occasionally and not compliant with EN50160	Yes, compliant with EN50160
Outages	Implemented but not used	On event occurrence

2G smart meters use Home Area Network (HAN) or in-home devices (IHD) to provide insights on electricity consumption and suggest actions to decrease cost. The two-way communication, between smart meters and HANs or IHDs, enables the exchange of messages for performing certain operations and/or retrieving information.

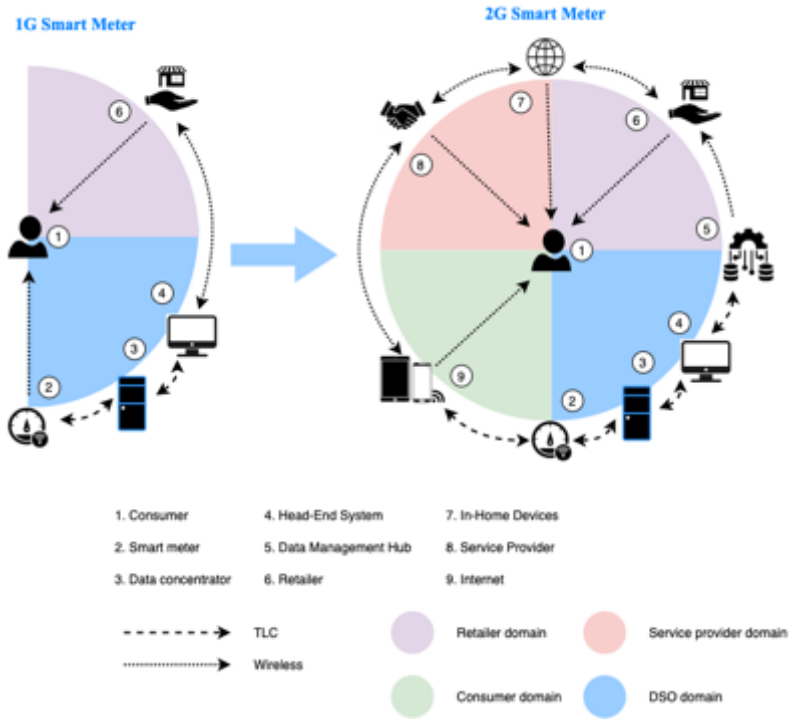

Figure 7: Smart meter evolution [35].

Figure 7 above show that the evolutions of smart meters create a whole new spectrum of domains, completing the circle around a satisfied consumer. Retailers could replace the fixed tariffs with flexible real-time pricing models. Data management/analysis allow DSOs to inform their consumers proactively in case of outages or even ask for

a demand response request in exchange for a reimbursement. Service providers have the opportunity to build useful applications to provide better insights to their consumers.

2.3.4. Smart meter roll-out in Europe

Smart meters play a major role in the digitalization of the conventional grid. European Union published its first provision of a potential smart meter roll-out back in 2009, the so-called Third Energy Package. Such provisions targeted a minimum of 80% roll-out by the year 2020 [39]. A necessary cost-benefit analysis (CBA) was conducted by the union members and supervised by Joint Research Centre of European Commission (JRC). The conclusion for a wide-scale roll-out was positive for most of the countries except Belgium, Czech Republic, Germany, Hungary, Ireland, Lithuania, Slovakia, Spain.

Table 2: Status of smart metering roll-out [36].

Country	**Roll-out according** [40]	**Roll-out according to** [41]
Austria	29%	11.8%
Croatia	n.a.	2.3%
Denmark	80%	69.1%
Estonia	100%	98.9%
Finland	100%	99.8%
France	80%	22.2%
Greece	37%	2.6%
Italy	100%	98.5%
Latvia	50%	36.3%
Luxemburg	80%	25.2%
Malta	80%	97%
Netherlands	85.2%	46.5%
Poland	11.5%	8.3%
Portugal	50%	25%
Romania	12%	4.8%
Slovenia	80%	58.2%
Sweden	90%	100%
United Kingdom	28%	19.9%

Observing Table 2, we could conclude that the expected objective of 80% of electricity consumers equipped with smart meters by 2020 is not achieved. Roll-out percentages reported by ACER report [40] in most cases are higher that the respective ones reported by EU-28 report [41]. The main reason for this is that ACER's report included smart meter installations in the industry. For instance, in Greece most industries are equipped with smart meters but the roll-out to residential consumers will start in 2022.

2.4. Advanced Metering Infrastructure (AMI)

The transition from a conventional grid to a smart one was/is not easy, functions that before were performed manually or weren't even possible had to be automated; electricity usage remote measurement, service connection/disconnection, tampering detection, voltage monitoring, outages identification/isolation, utilities - customer communication. Hence, an integrated system of smart meters, communications networks, and data management systems was designed to enable two-way communication. The name of such a system is Advanced metering infrastructure (AMI).

2.4.1. Definition

As per Gartner [42] AMI is defined as: "Advanced metering infrastructure (AMI) is a composite technology composed of several elements: consumption meters, a two-way communications channel, and a data repository (meter data management). Jointly, they support all phases of the meter data life cycle — from data acquisition to final provisioning of energy consumption information to end customers (for example, for load profile presentment) or an IT application (such as revenue protection, demand response or outage management)."

2.4.2. AMI vs AMR

Before commencing the discussion on details of AMI, it is important to differentiate two terms, Automatic meter reading (AMR) and AMI. AMR is an improvement of the conventional energy meter where the data collection, recording and billing is done manually. However, this process is unidirectional, energy meter responds to the device, but the device does not respond back. AMI on the other side, offers two-way communication between utility and metering end points. Energy management and energy conservation have become very important especially in times where the energy prices skyrocket. AMI, unlike AMR, proved to be a powerful tool for helping consumers improve their energy habits by recording energy patterns and informing consumers timely about their budgeting and billing, so energy wastage could be avoided [43].

2.4.3. High level Overview

Considering the definition above, AMI is not a single technology; rather, it is an infrastructure/ecosystem integrating smart meters, communication networks and management systems. A very important aspect is that AMI enables two-way communication between consumer, smart meter, and distribution system operator (DSO) [44]. The two-way communication facilitates operations that were nearly impossible to fulfill before AMI; rapid detection, diagnosis, and resolution of power quality problems, automatic notification of outages and self-healing capabilities, energy consumption pattern identification detecting possible energy theft or tampering.

Figure 8 depicts a high-level overview of an AMI system. Such a system consists of smart meters, transmitters, and a Meter Data Management System (MDMS) [45]. Smart meters collect time-based data that are transmitted through available fixed networks such as Power Line Communications (PLC), Broadband over Power Line (BPL), Fixed Radia Frequency, as well as public networks like cellular, landline. The communication equipment receives these data and forwards them to operator's data center. An important security aspect is to keep customer's usage data encrypted. The heart of operator's data center should be the MDMS where data are being analyzed for producing accurate bill, monitor electricity system performance and discovering insights. A subset of usage data is being available in a refined way to the customer so they could manage electricity usage and cost.

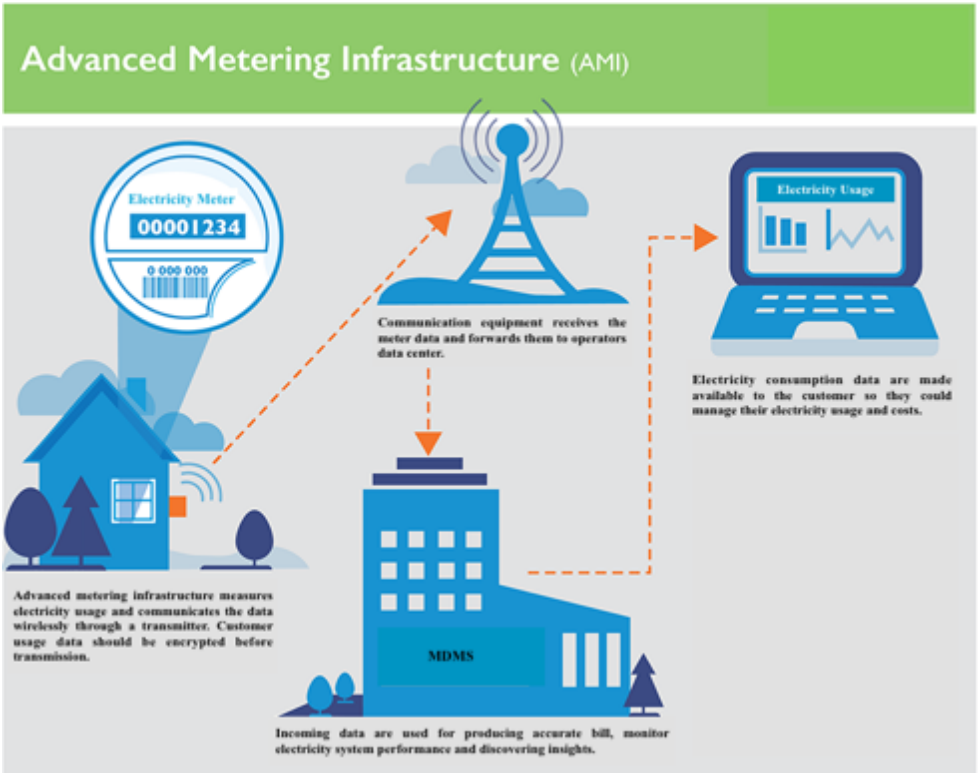

Figure 8: Advanced Metering Infrastructure (AMI).

2.4.4. Detailed Overview

There are two key components in smart meters, meter board and communication board, connected using a serial port (see Figure 8). Meter board is responsible for measuring power consumption and storing in its set of tables sensitive information, keys, and passwords, required for a secure communication. The communication board gathers data/information from meter board in order to perform the communication with external nodes such as collectors and home appliances [46].

Figure 9 provides a more detailed overview of AMI system structure and the integrated technologies. Let's describe the component of such a system:

1) User Side:
   a) Smart Meters (SMs) [47]. A smart meter is an electronic device that records near real-time information and reports them in short intervals (e.g., every 5 mins) to the DSO. Such information could be electricity consumption, voltage levels, power factor and current. The enabled two-way communication provides on one side consumers with clarity and enhanced consumption profiles and on the other side DSOs with better monitoring capabilities and accurate billing. High-resolution data from smart meters provide rich information on the electricity consumption behaviors and lifestyles of the consumers [48].
   b) Distributed Energy Resources (DERs) [49]. Distributed energy resources are small scale, modular in same cases renewable in many, energy generation and storage systems. Typically, their production does not exceed 10 megawatts (MV). Examples of DERs are wind turbines, photovoltaics (PV), microturbines etc.
   c) Gateways (GWs) [50]. As in digital networks, gateways are responsible for the conversion of protocol and communication between two heterogeneous networks, e.g., home are network, wide area network.
2) Wide Area Communication Infrastructure [51]. Acts like a bridge between user side and DSO enabling a bidirectional communication. Cellular networks and power line communication system are examples of communication medians.
3) Management side:
   a) Meter Data Management System (MDMS) [52]. MDMS is a system that handles storage, management, and analysis of metering data. It comprises of tools that facilitate the. Interaction between Distribution Management System (DMS), Outage Management System (OMS), Consumer Information System (CIS) and Geographic Information System (GIS).
   b) Demand Response (DR) [53]. Monitoring power flows is highly important since it enables DSOs to react in time on variations in consumption levels, this results in efficiency in investment on power generation,

transmission, and distribution assets. Another direct benefit is an accurate real-time pricing policy. Demand Response would enable loads to be controlled in response to supply side (generation) availability and associated tariffs.

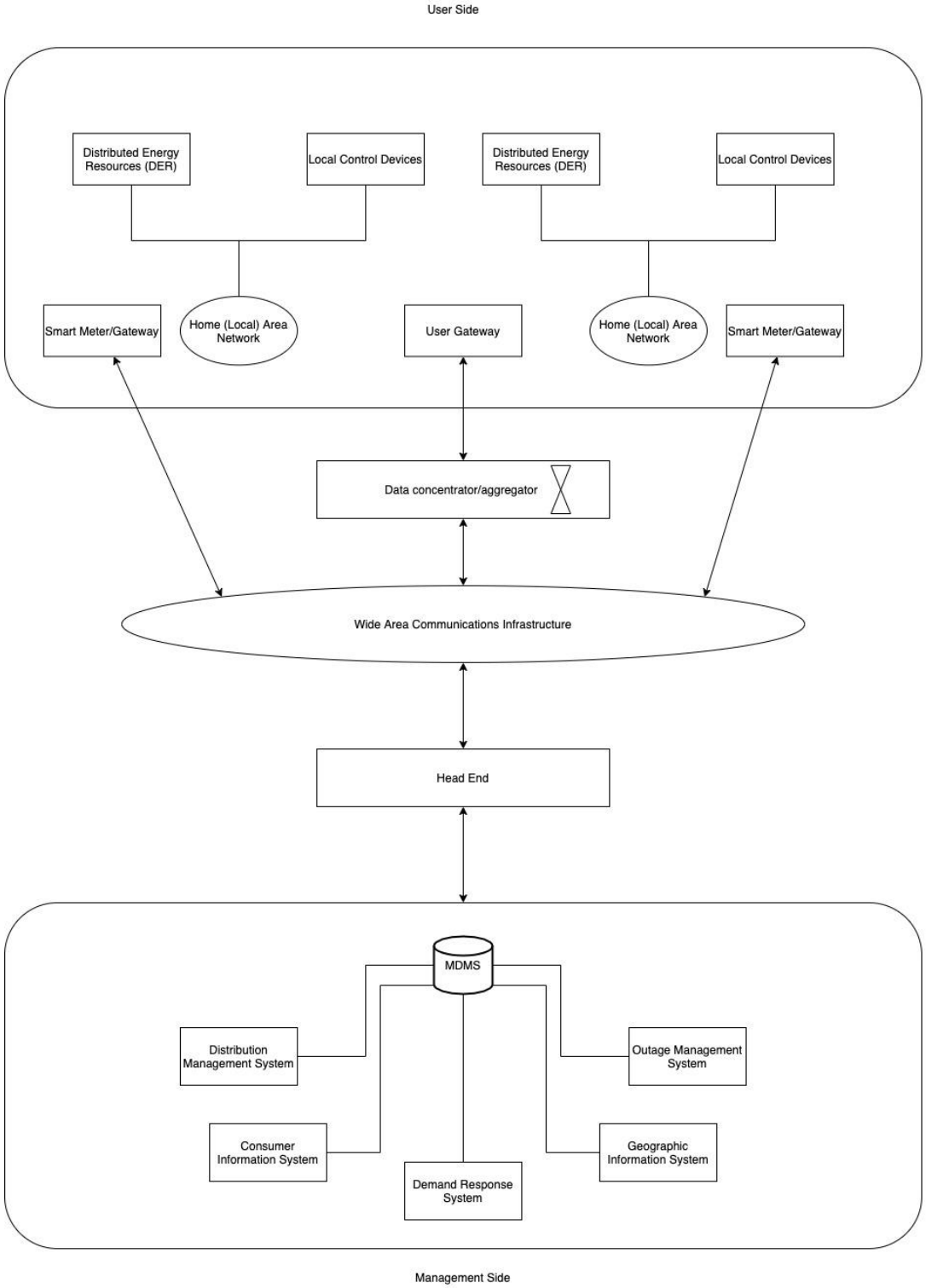

Figure 9: AMI structure & Integrated technologies.

From communication perspective, AMI makes use of the following networks:

- Home Area Network (HAN) [54]. This is a small-scale network (within home premises) that connects smart devices and smart meters. Wireless technologies such as 802.11 protocol, ZigBee and Home-Plug

are the most common used for building HANs. Lightweight security mechanisms are deployed in the smart meters as they are generally resource constrained [55].

- Neighborhood Area Network (NAN) [56]. It is formed by combining HANs. Power consumption and security alarm data are being transmitted for enhanced energy management.
- Wide Area Network (WAN) [56]. A data concentrator/aggregator collects the data from a group of smart meters and then it sends them to the Head End (see Figure 9) through the Wide Area Communication Infrastructure. Hence the main task of such a network is to connect the local network to the Head End.
- Smart Meter Gateway (SMGW) [57]. This gateway acts as an intermediate between a group of smart meters and the WAN. Its main task is to provide authentication and aggregation of meter messages.

2.4.5. Security Challenges

The development of such digital infrastructures brings along security concerns/challenges [58].
First let's summarize the general security requirements [46]:

- **Confidentiality**. It is of utmost importance ensuring the privacy of consumer's information/data. Tampering of smart meter to access illegally the stored data or the unauthorized access to such data should be prevented [59].
- **Integrity**. Integrity is perceived as the mechanism of preventing alternations of the data in the process of bidirectional communication between the smart meter and DSO's data center [60]. AMI should be robust against hackers who launch attacks by impersonating authorized entities.
- **Availability**. User data should be collected within configured time intervals and control command should be delivered in time. Scenarios including component failures and communication failures due to network traffic, interference, band-width loss, degeneration should be avoided by any means [46].
- **Accountability**. It refers to the fact that data receivers will not deny receiving of data and vice versa [46]. Timestamping of data messages is therefore necessary to ensure accountability. Audit logs are the most common way of accountability maintenance.

These security challenges can be categorized in three main topics: end user privacy, resilience against cyber-attacks and power theft [46].

1) End User Privacy. User data are transmitted through wired/wireless networks for storage and further analysis, hence they become vulnerable to data theft and/or manipulation [37]. These data are possible to be reversed engineered [61] and derive critical information regarding consumer's profiles, number of occupants in a household, time of occupancy, existence of electric vehicle, existence of alarm or security system etc.
2) Resilience Against Cyber Attacks. Digitalizing the energy grid (smart grid) bring along the vulnerabilities of such networks in cyber-attacks. Attackers will try to hack the grid network and destabilize it. Smart meters should have a secure connection with the network and even if a smart meter is compromised it shouldn't be possible to obtain critical information of other meters or even worse gain access to the network.
3) Power Theft. Losses in energy can be occurred in any stage of the power flow, generation, transmission, distribution, utilization. Traditional (existing) systems use electro-mechanical meters which are easily manipulated. The main way to detect power theft is the direct connection to distribution lines and grounding the neutral wire [62]. The use of smart meters will almost eliminate this problem.

2.5. Load Forecasting

Forecasting is an estimation of uncertain future events, which could be used to improve decision making and planning. Knowing in advance a possible outcome; even containing some error rate, allows to better manage expectations and avoid potential risks. Load forecasting is related to energy sector; Gartner [63] provides the following sophisticated definition:

"*Load forecasting minimizes utility risk by predicting future consumption of commodities transmitted or delivered by the utility. Techniques include price elasticity, weather and demand response/load analysis, and renewable generation predictive modeling. Forecasts must use regional customer load data, with time series customer load profiles. Accurate forecasts require adjustments for seasonality. Distribution load forecasting must be reconciled with distribution network configuration as part of the distribution circuit load measurements*."

Let's try to deconstruct a bit the definition above by presenting an overview of supply-demand equilibrium, the forecasting horizons, the affected stakeholders, and the factors affecting load forecasting.

2.5.1. Supply – demand equilibrium

One of the main challenges of system operators is to keep the balance between supply and demand; provide the amount of energy demanded and no more or less than that. Supply higher than demand leads to energy waste and in many cases, penalties are imposed. Supply lower that demand has as a result incapability of serving all consumers and might lead to blackouts. It is obvious that load forecasting is playing a very important role as a key operation of power grid planning and future evolvement [3].

However, reliability, security and optimization of grid operations become more and more complicated with the penetration of renewable and distributed generation sources. The dynamic nature of those resources makes it harder for the utilities to forecast how much energy they need to cover the demand [64]. As a result, there is a potential money and resource waste for generating/purchasing power that is not needed. Predicting energy consumption allows the utilities to plan ahead and optimize their resource planning and future energy generation. An accurate load forecast is of significant important due to the large amount of money involved in energy budgets [65].

2.5.2. Forecast Horizons

Based on the lead time, load forecasting has four main horizons: very short-term load forecasting (VSTLF), short-term load forecasting (STLF), medium-term load forecasting (MTLF) and long-term load forecasting (LTLF) [3]. VSTLF is used for near real-time control and it's prediction horizon spans from minutes to 1h ahead. STLF is used for the day-to-day operations of the utilities such as optimal scheduling on energy generation and transmission and its horizon spans from 1h to 7 days. MTLF's horizon ranges from 1 week to 1 year. Its main purpose is for forecasting fuel purchase, maintenance, utility assessments. At last, but not least, LTLF forecasts beyond a year and up to 20 years ahead. It is suitable for strategic planning, new generations, long term changes in infrastructure or in economic model [64].

2.5.3. Affected Stakeholders

The following stakeholders could benefit from accurate load forecasting [66]:

- Transmission System Operators (TSOs), DSOs, grid operators. Awareness of future load transmission/distribution requirements helps on avoiding grid hotspots and congestions, investment planning, grid reinforcements, and optimal load distribution.
- Energy retailers. As the market is decentralized and becomes more open, load forecasting is of utmost importance since retailers have to build sustainable short-, mid- and long-term business models to survive in such a competitive environment.
- Energy market participants. Their main goal is to buy or sell energy or derivatives on energy. Forecasting load fluctuations helps them ameliorating their transaction strategy.
- End-users. Load forecasting helps retailers secure more competitive prices on stock market; hence the consumers benefit from lower prices. On top, awareness of forecasted consumption gives consumers the flexibility to adjust their energy consumption habits accordingly.

2.5.4. Factors affecting load forecasting

The total load of the system consists at all times of the sum of the demand of individual consumers, who depending on the behavior of their electricity demand could be divided into industrial, commercial, agricultural, domestic, etc. The electricity consumed by them is affected from a number of different and in many cases indeterminate factors, which can be summarized in the following four categories [67]:

- Financial factors. The economic environment of a household directly affects the demand and usage of energy. Such a household uses a variety of electric appliances or EVs, which in most cases are of a higher energy class; hence energy saving is achieved. Financial factors do not affect STLF since the financial profile of a household does not change in such a time span. Their impact, however, is particularly important in the long run.
- Chronological factors. There are three main chronological factors; seasonal changes, weekly/daily cycle, national holidays, and religious fests. Seasonal changes are more bind to seasonal variables like temperature and hours of sunshine. These variables strongly influence the use of heating or air conditioning appliances, at the same time they affect human activity. For instance, during summer holidays there is a decrease in demand in urban centers and an increase in tourist resorts. Other important seasonal factors are the change of time (winter/summer) and the start of the school year. The weekly/daily periodicity of the load is a result of the periodicity of the work-rest cycle of the population. Weekdays present a different load pattern than weekends. The load profile is also different in national holidays and festive periods.
- Weather. Weather conditions cause significant changes in the electricity consumption. This is because the operation of many electrical appliances (heating, air conditioning) depends on weather conditions. Humidity, rainfall, wind, and sunshine affect demand. However, the main role in the consumption of electricity is played by the temperature, which is a basic condition for a satisfactory load forecast in both the short and medium term. In fact, for systems that cover a large geographical area, it is necessary to take into account the temperatures in different areas in order to calculate the exact effect on the load.
- Ad-hoc. Such factors are certain events - such as large strikes, elections, special programs on television - which, although known in advance, are difficult to assess their impact on demand. Some other events like a pandemic are not even possible to predict. Finally, we should mention factors such as the development prospects of an area, and the growth rate of the population, which affect the long-term consumption of electricity and are characterized by great uncertainty.
- demographics

3. Literature Review

As technology advances and societies develop the electricity supply plays a crucial role in economic activities and daily life. The power network needs to keep so it has been transformed to a smart grid where the power generation, transmission and utilization processes are empowered with ICT technologies. The increased amount of collected data refined load forecasting contributing the maximum to a transition toward an intelligent power system.

Load forecasting can be divided into three categories considering the forecast step size: long-term, mid-term and short-term. LTLF predicts on a horizon of a year or more and it is crucial for electricity infrastructure construction/development planning [68]. MTLF ranging from one week to several months and one of its main goals is to develop an efficient fuel supply plan. STLF on the other hand has a very short forward projection time frame; hours to a small number of days and plays an important role in energy demand management and day-to-day operations, e.g., energy trading [69]. In this work we are going to focus on STFL for residential users.

Since productions methods and consumers lifestyle diversifies more and more, accurate STLF on residential level is of great importance for power system operation. Smart grids create opportunities but bring challenges along, utilities are exploring ways to manipulate commercial and residential loads in near real-time [70]. However, integration of intermittent renewable electricity generation rises the complexity of such function since renewable sources such as sun or wind are highly volatile. Accurate forecasts of those non controllable electrical loads are necessary for an efficient DR. High accuracy of energy demand real-time prediction helps utilities broadcasting signals to consumers to take actions to sustain the demand – supply equilibrium. If the forecast model over-predicts the redundant energy requires the use of expensive storage or in the worst-case scenario will lead to energy waste. On the other hand, if the model under-predicts the demand, utilities need to spin-up expensive quick-response generation mechanisms so they can compensate the supply shortfall.

STLF desires to maintain supply quality and attain cost reduction by anticipating demand fluctuations and weather, especially when renewable sources are being integrated into the electricity generation mix. Extreme demand peaks could be avoided by using load shifting and load scheduling applications. Utilities on their side use STFL output to create a more efficient power generation plan; spin-up production sources e.g., generators that require time to produce output. Additionally, in case of a multi-generation system, STLF can be used to develop a schedule for the different sources [71].

Residential electricity load is affected by weather, electricity price, lifestyle, holidays etc., something that makes STLF on residential level difficult and challenging. STLF predicts future load trends by ingesting knowledge of current and historic data. Numerous factors have to be considered to lead to an accurate forecast: weather forecast, time of the day, type of day (workday/holiday), historical demand trends. Classical modeling techniques have tough times considering the impact of all these factors; hence Artificial Intelligence (AI) techniques have become more and more popular.

Traditional statistical models such as regression analysis [72], moving average [73], exponential smoothing [74], and stochastic time series models [75] are applied for time series forecasting. Time series decomposition model [76] investigates the factors that affect load forecasting and notifies that this approach might ignore the correlation between time periods; hence might lead to biases forecasting. Another work [77] proposes a lifting wavelets method, which removes noise from historic data and performs nonlinear feature extraction. Researchers have used models like ARIMA, SARIMA and, SARIMAX [78][79][80]. However, the evolution of machine learning technologies and available data retrieved from smart meters shifted the focus to computational models.

The development of AMI, hence the integration of smart meters into the grid boosted the interest of researchers on short-term load forecasting on residential level. One of the first work on the area [81] proved that the near real-time data gathered by AMR technology greatly improves accuracy but at the same time increases computational complexity. In recent years, the evolution of deep learning models made them very appealing and effective on short-term load forecasting. A hybrid model is proposed in [82], combining an extreme learning machine and an extended Kalman filter for online short-term load prediction. The dataset in this study is relatively small, including hourly consumption data of two residential and two commercial buildings. The lack of data is not an ally when using deep learning methods. Traditional recurrent neural network (RNN) has the problem of gradient disappearance and long-term dependence [83], therefore the focus shifted more to LSTMs. In [84], the proposed framework based on LSTM [85] recurrent network, forecasts the load of 69 consumers. In addition, the study performs benchmarking (mean absolute percentage error, MAPE) between the proposed model and various state-

of-the-art techniques. However, the researchers pre-screened the users, which might have affected the results. A deep-based conditional probability density function was used in [86]. The model achieved high accuracy in both single households and aggregated load of 3500 residential houses. Studies [87] and [88] proved the positive impact of weather load forecasting accuracy. The cross-correlation between load and several weather factors has been examined to select the factors with the strongest correlation. Both studies used LSTM model for the short-term load forecasting on residential consumers. Work [89] results in high accuracy and generalization ability, using a stacked self-encoder model. Self-encoders are unsupervised deep learning models have strong feature extraction capabilities. Seq2Seq approach based on RNN [90] demonstrated promising results in load forecasting for a commercial and a residential consumer. Another work [91] performed in one residential consumer supported the efficiency of such an approach.

3.1. Theoretical Background

3.1.1. Time Series Analysis

Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. The main goal is to identify the effect of one variable on either itself or another over time, in order to get insights regarding the nature of a problem or/and create predictive models.

The set of data which is collected over time and expresses the evolution of the values of a variable during equal time periods, is called a timeseries or sequence. By a timeseries we usually mean a sequence of observations $\{X_t: t = 0,1,2,\cdots,T\}$ where X_t expresses the state of a system at time t. According to the mathematical definition, a time series is defined as the set of observations $x_1, x_2, \cdots, x_n$ of the values $X_1, X_2, \cdots, X_n$ of a random variable X at equidistant moments in time $t_1, t_2, \cdots, t_n$. Therefore, this sequence of random variables is called a stochastic process and is denoted as $X(t)$ [91].

For the statistical processing of timeseries it is particularly useful to distinguish its four components [93]:

- **Trend**: The long-term movement, upwards or downwards, of a timeseries when observed over an extended period of time, could be defined as trend. The trend does not exist when its movement is parallel to the axis of time, without fluctuations. The most common methods of determining trend are moving average and least squares.
- **Seasonality**: A timeseries exhibits seasonality when its dispersion exhibits the same behavior over time periods t. Usually periodic fluctuations refer to time intervals shorter than a year. An example of seasonality is the increase on energy consumption during the winter.
- **Cyclicity**: Cyclicity expresses the cyclical fluctuations for periods longer than a year that are repeated at equal time intervals and are due to external factors.
- **Irregular Fluctuations / Outliers**: They are values that are significantly different from the rest of the observations, where they are usually due to some unpredictable factor and create problems in modeling.

3.1.2. SARIMAX

Autoregressive and moving average models work with stationary and linear data. However, in many cases, the data is non-stationary. Autoregressive Integrated Moving Average (ARIMA) models are used to deal with non-stationary data. An ARIMA model consists of three parts, namely autoregressive (AR) terms, moving average (MA) terms and differencing operations (I). The differencing operation is used to create a stationary series for modelling [94]. In this operation, a value is replaced with the difference of the value and its previous value [94]. A generalized form of the ARIMA model known as the Seasonal ARIMA (SARIMA) is used to handle seasonality in data. This class of ARIMA models deals explicitly with seasonality in data by using seasonal AR, MA, and differencing terms in the model. External variables can also be added to the model through an exogenous regressor term. Seasonal ARIMA with exogenous regressors (SARIMAX) enables the user to add the effects of external variables to the model. Exogenous variables are defined as variables that influence a model but are not influenced by it. The weather is considered an exogenous variable in the context of an energy consumption model of a building.

The SARIMA model is defined as:

$$SARIMA(p,d,q) \times (P,D,Q)_s \tag{3.1.2-1}$$

Where:

Trend Elements are:

- p: Autoregressive order (AR).
- d: Difference order (I).
- q: Moving average order (MA).

Seasonal Elements are:

- P: Seasonal autoregressive order (SAR).
- D: Seasonal difference order. D=1 would calculate a first order seasonal difference (SI).
- Q: Seasonal moving average order. Q=1 would use a first order errors in the model (SMA).
- s: Single seasonal period.

Exogenous variables

- X: exogenous variable/features. Exogenous variables like weather data and the hour of day have been used in this research. Those have been picked up because they have a strong correlation with consumption.

A SARIMAX is mathematically represented as [94]:

$$y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \cdots + \beta_k X_{k,t} + \frac{\left(1 - \theta_1 \mathrm{B} - \theta_2 \mathrm{B}^2 - \cdots - \theta_q \mathrm{B}^q\right)\left(1 - \Theta_1 \mathrm{B}^s - \Theta_2 \mathrm{B}^{2s} - \cdots - \Theta_Q \mathrm{B}^{Qs}\right)}{\left(1 - \varphi_1 \mathrm{B} - \varphi_2 \mathrm{B}^2 - \cdots - \varphi_p \mathrm{B}^p\right)\left(1 - \Phi_1 \mathrm{B}^s - \Phi_2 \mathrm{B}^{2s} - \cdots - \Phi_P \mathrm{B}^{Ps}\right)} Z_t \quad (3.1.2\text{-}2)$$

where:

- y_t denotes the value of the series at time t.
- $\mathrm{X}_{1,t}, \mathrm{X}_{2,t}, \cdots, \mathrm{X}_{k,t}$ denote observations of the exogenous variables.
- $\beta_0, \beta_1, \cdots, \beta_k$ denote the parameters of the regression part.
- $\varphi_1, \varphi_2, \cdots, \varphi_p$ denote the weight of the nonseasonal autoregressive terms.
- $\Phi_1, \Phi_2, \cdots, \Phi_P$ denote the weight of the seasonal autoregressive terms.
- $\theta_1, \theta_2, \cdots, \theta_q$ denote the weight of the nonseasonal moving average terms.
- $\Theta_1, \Theta_2, \cdots, \Theta_Q$ denote the weight of the seasonal moving average terms.
- B^s denotes the backshift operator such that $B^s y_t = y_{t-s}$.
- Z_t denotes the white noise terms.

3.1.3. Box-Jenkins Method

The Box-Jenkins methodology is one of the most adopted forecasting methods using ARIMA models and is applicable to several domains. According to Box and Jenkins [95], the modelling process is broken down into three iterative steps, namely identification, estimation, and diagnostic checking.

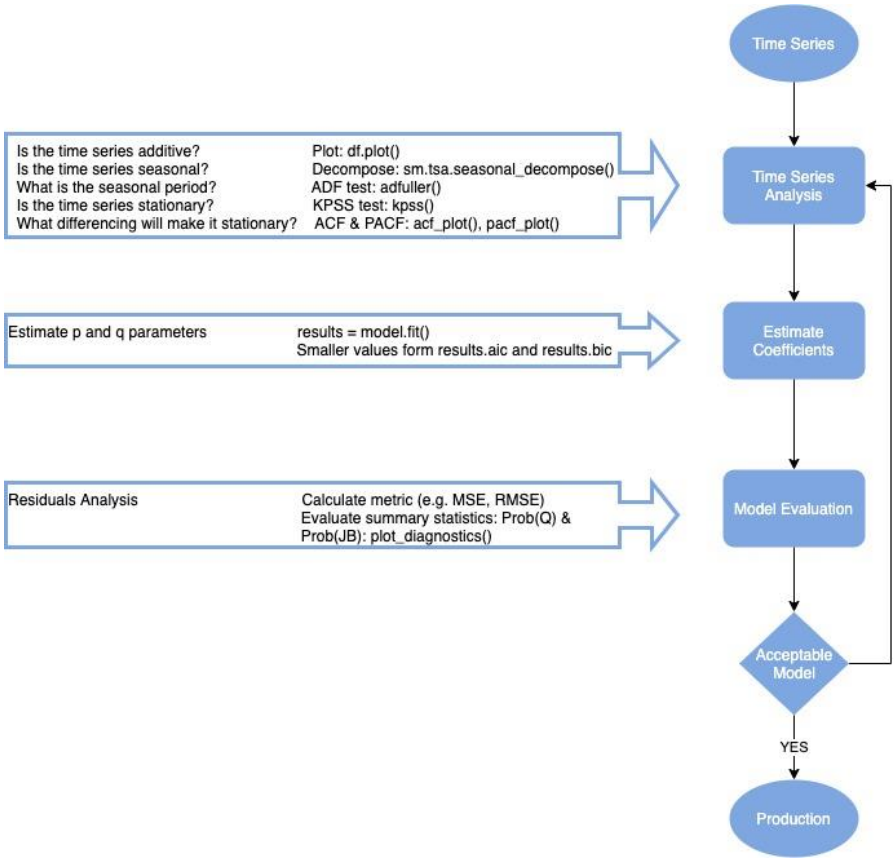

Figure 10: Box-Jenkins method.

Identification is the process where data and all related information are used to select a subclass of models that best suit the data. Initially, we need to assess whether the time series is stationary or not using unit root statistical tests. If not, we need to convert it stationary by differencing. Differencing is a technique that attempts to increase stationarity by subtracting a previous observation from the current observation. Subtracting the observation immediately preceding the current observation produces a first difference. We should avoid over differencing since it will result in extra serial correlation and complexity. Another part of identification process is to configure Autoregression (AR) and Moving Average (MA) models. AR is a model that uses the dependent relationship between an observation and some number of lagged observations. MA is model that uses the dependency between an observation and residual errors from a moving average model applied to lagged observations. The number of lag observations included in the model, also called the lag order p and the size of the moving average window, also called the order of moving average q, are derived with the help of two diagnostic plots Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF).

- ACF plot summarizes the correlation of an observation with lag values. The x-axis shows the lag and the y-axis shows the correlation coefficient between -1 and 1 for negative and positive correlation.
- PACF plot summarizes the correlations for an observation with lag values that is not accounted for by prior lagged observations.

Some useful patterns you may observe on these plots are:

- The model is AR if the ACF trails off after a lag and has a hard cut-off in the PACF after a lag. This lag is taken as the value for p.
- The model is MA if the PACF trails off after a lag and has a hard cut-off in the ACF after the lag. This lag value is taken as the value for q.
- The model is a mix of AR and MA if both the ACF and PACF trail off.

Estimation is the process to train the model using different parameters and choose the best model based on some criteria. The most common criterion for model selection is Akaike's Information Criterion (AIC) [33]. AIC essentially measures how well it fits the data, while penalizing complexity. Therefore, AIC reduces the risk of both overfitting and underfitting. A model that fits the data well and uses many predictors will have a larger AIC compared to a model that has the same goodness of fit but uses fewer predictors. Therefore, when comparing models, the one with the least AIC is chosen as the winner. It should be emphasized that the AIC of a model is a relative measure and is meaningful when compared to other models.

Diagnostic checking is the process of finding evidence that the model is not a good fit for the data. The first check is to check whether the model overfits the data. Generally, this means that the model is more complex than it needs to be and captures random noise in the training data. This is a problem for time series forecasting because it negatively impacts the ability of the model to generalize, resulting in poor forecast performance on out of sample data. Careful attention must be paid to both in-sample and out-of-sample performance and this requires the careful design of a robust test harness for evaluating models. Residuals errors, a review of the distribution of errors can help tease out bias in the model. The errors from an ideal model would resemble white noise, that is a Gaussian distribution with a mean of zero and a symmetrical variance. For this, you may use density plots, histograms, and Q-Q plots that compare the distribution of errors to the expected distribution. A non-Gaussian distribution may suggest an opportunity for data pre-processing. A skew in the distribution or a non-zero mean may suggest a bias in forecasts that may be correct.

3.1.4. LSTM

A Recurrent Neural Network (RNN) [96] is a special type of artificial neural network that handles time series data or sequences effectively. Feed forward neural networks have hard time on these problems since in sequences the present data point depends in the previous one. RNNs trying to solve this problem by introducing the concept of memory, hence storing the states of previous inputs to generate the next output. However, RNNs suffer from the problem of long-term dependencies [97]. Sometimes we only need to look up recent information to predict the outcome; if for instance we are trying to predict the next work in phrase "the apple tree grows …", we won't need further context come up with the word apple. In such cases the gap between the relevant information and the intended outcome is small. The more this gap grows the harder is to accurately predict the next output.

Long Short-Term Memory (LSTM) network was introduced by [98] and is a temporal cyclic neural network [85], which solved the problem described above. LSTM replaces the hidden layer neurons of a RNN with memory units. The main idea is to delete invalid information and/or retain important information as the time series progresses. The ability to identify and remember temporal correlations makes these networks ideal for applications such as speech recognition and language translation. Electricity consumption of individual households is based on consumers behavior, and it is quite ad-hoc. LSTM is designed to extract a consumption pattern, then keep it in its memory and finally make the forecast [84].

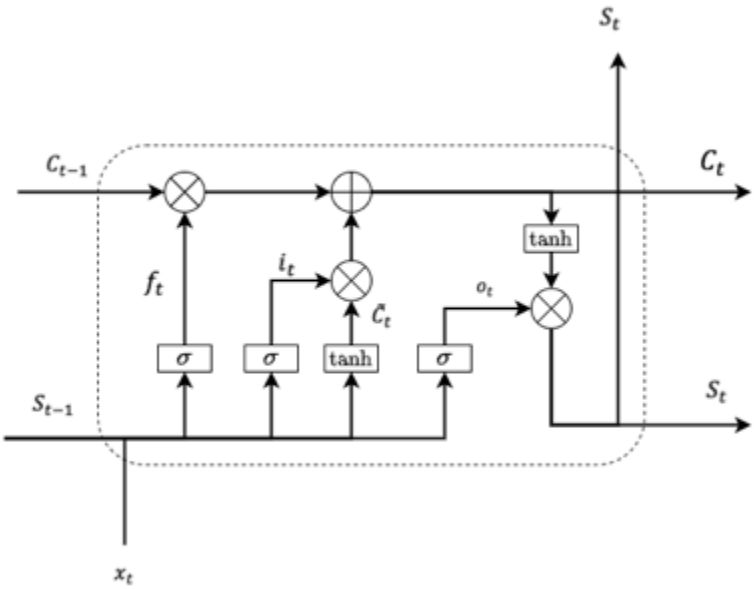

Figure 11: LSTM block structure

The structure of an LSTM cell block is shown in Figure 11. Such block is consisted of a memory cell, an input gate, an output gate and a forget gate. The memory cell is responsible of remembering the previous state and keeping also track of the correlation between the elements in the input sequence.

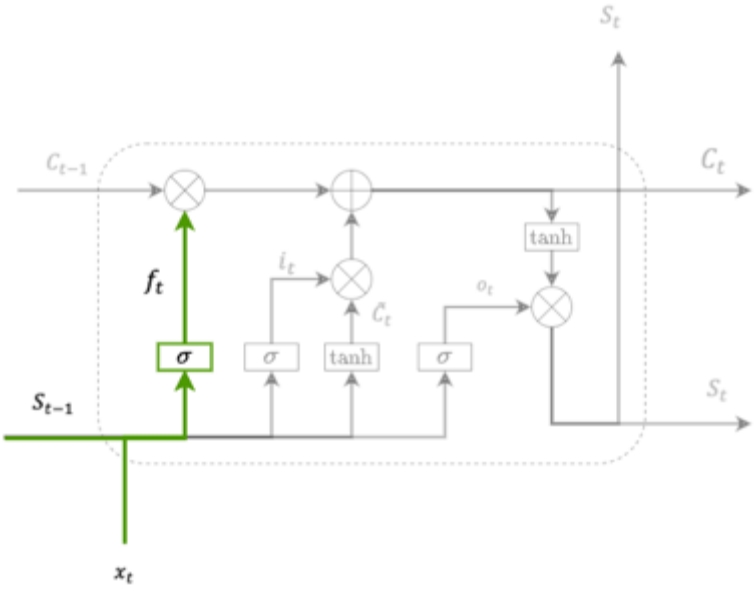

Figure 12: LSTM walkthrough step 1.

Initially, Figure 12, the LSTM network needs to decide what information is redundant and needs to be thrown away. This is achieved by a sigmoid layer called forget gate layer. It combines the current input x_t with the previous cell state C_{t-1} and outputs a number between 0 and 1. A 0 means do not keep this while a 1 means keep this. The formula of the forget gate is defined as:

$$f_t = \sigma\left(W_f[s_{t-1}, x_t] + b_f\right) \tag{3.1.4-1}$$

Where W_f is the weight matrix and b_f is the bias of the forget gate.

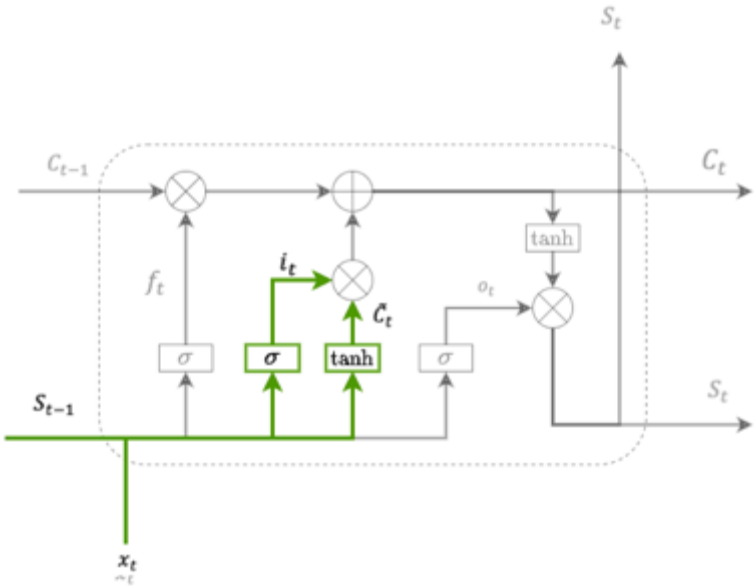

Figure 13: LSTM walkthrough step 2.

The next step, Figure 13, decides what new information needs to be stored in the cell state. First the input gate layer, sigmoid layer, decides which values needs to be updated. Next a $tanh$ layer constructs a vector of new candidate values, $\tilde{C}_t$. There are two formulas describing the two parts:

$$i_t = \sigma(W_i[s_{t-1}, x_t] + b_i) \tag{3.1.4-2}$$

$$\tilde{C}_t = tanh(W_c[s_{t-1}, x_t] + b_c) \tag{3.1.4-3}$$

After combining the two parts calculated above, the actual update needs to be performed; update the old cell state C_{t-1} in the new one C_t.

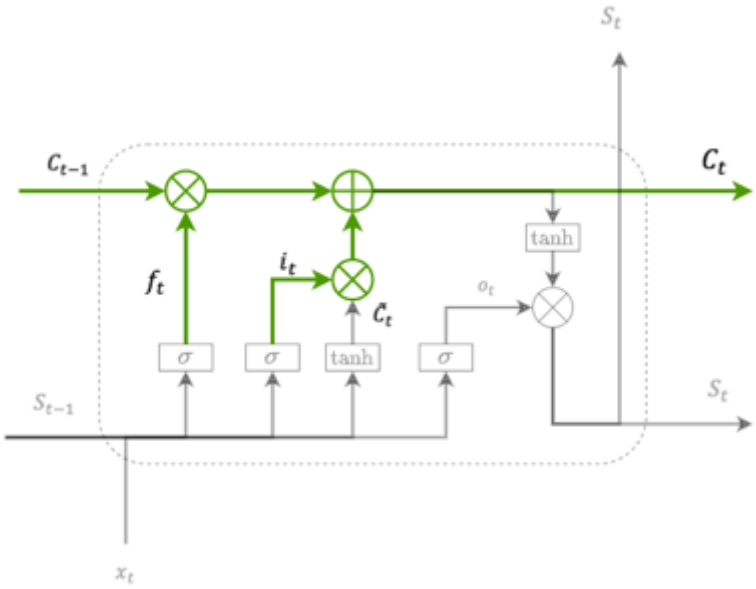

Figure 14: LSTM walkthrough step 3.

As shown in Figure 14, the old state C_{t-1} is multiplied by the forget gate f_t. Then we add the new candidate values, scaled by the updated of each state value; described by the following formula.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.1.4-4}$$

Finally, Figure 15, the output will be based on a filtered version of our cell state. A sigmoid layer is applied to determine the parts of the cell state that are going to be outputted. Thereafter the cell state is passed through a $tanh$ to limit values between -1 and 1 and multiply it by the output of the sigmoid gate, o_t. The reason of doing this is to output only the parts we have decided.

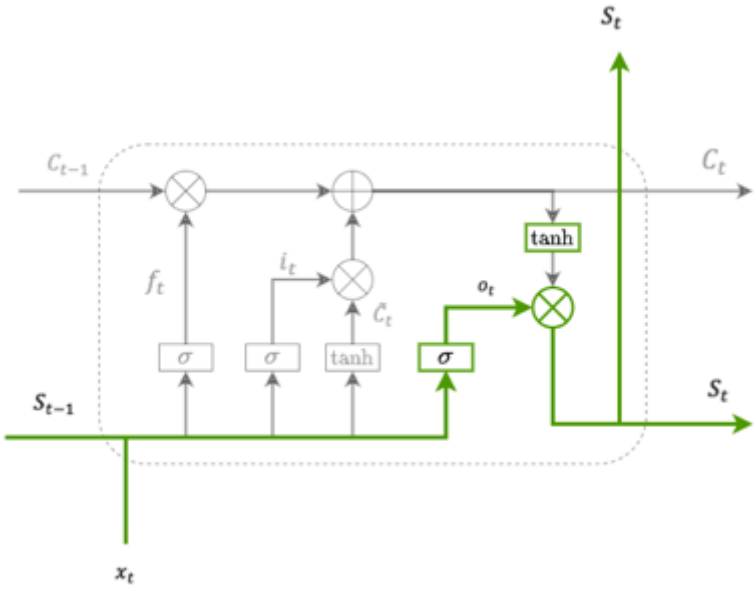

Figure 15: LSTM walkthrough step 4.

The formulas related to this step are the following:

$$o_t = \sigma(W_o[s_{t-1}, x_t] + b_o) \tag{3.1.4-5}$$

$$S_t = o_t * tanh(C_t) \tag{3.1.4-6}$$

4. Methodology

4.1. Problem definition

The idea is that a device (agent) is connected to the smart meter of a residential house, which makes an accurate load forecast of the household and adapts to changing conditions in energy use. The forecast has a 24 hour horizon considering the consumption of the past 24 hours plus the weather forecast. 24 hours after the forecast is made, an evaluation takes place of how well the forecasts of IMPORT_KW corresponded to reality.

Smart meter data fall under the time series category, more specifically in our case they fall under the category or Multivariate Time Series (MTS). MTS has more than one time-dependent variable. Each variable depends not only on its past values but also has some dependency on other variables. This dependency is used for forecasting future values. Our dataset includes consumption data and it is combined with weather data, therefore there are multiple variables to be considered to optimally predict consumption.

4.2. GridFlexHeeten dataset

The data in this dataset was collected during the GridFlex Heeten project [99]. The data was collected between August 2018 and August 2020 in 77 households all situated in Heeten (The Netherlands) and consists of electricity consumption and gas usage per minute per household. All participating households specified their data could be used in further research and the data of this project was collected in accordance with a privacy-by-design approach.

4.2.1. Energy Consumption Data

The data in this dataset is subdivided as a value per house, per appliance-group, per measurement type, per time interval. These subdivisions/columns are explained further:

- "time":
  - YYYY-MM-DDThh:mm:ssZ: one-minute time interval (PT1m/YYYY-MM-DDThh:mm:ssZ) at the end of which the measurement value is taken, denoted in ISO 8061 (second precision), ranges from 2018-08-01T01:59:00+02:00 to 2020-08-31T23:58:00+02:00. Note that this date is denoted in local time, so CET (or CEST), taking Daylight Saving Time into account.
- "house":
  - HouseX: The household identifier, with X ranging from 1 to 77.
  - HouseTest: A dummy household to test if the connection is working.
- "appliance":
  - SMARTMETER: Contains the measurements related to the smart meter, so the complete household.
  - PVMETER: Contains the measurements related to the PV system (not available in all households).
  - BATTERY: Contains the measurements related to the battery system (not available in all households).
- "measurement":
  - BATTERY_EXPORT_KW: Total energy the battery discharged since it was connected to the system, in kWh.
  - BATTERY_IMPORT_KW: Total energy the battery charged since the battery was connected to the system, in kWh.
  - BATTERY_KW: Average power output of the battery in the last minute (negative means charging), in kW.
  - BATTERY_TARGET_KW: The requested power output of the battery, in kW.
  - BATTERY_TARGET_MODE: Optimization strategy of the battery, where 0 = local, so the battery tries to match BATTERY_TARGET_KW to BATTERY_KW, 1 = household, so the battery tries to steer BATTERY_KW such that TOTAL_KW matches BATTERY_TARGET_KW, 2 = failsafe, so the battery charges to a safe State of Charge (SoC) regardless of BATTERY_TARGET_KW.

- CHARGE_MODE: Indicates the mode of battery which dictates what the battery can do, where 0 = battery is idle, 1 = battery can only charge, 2 = battery can only discharge, 3 = battery can do both.
- CURRENT_PHASE_1: Household current on phase 1, in A. Value is very inaccurate, but can be used to identify the phase the household is connected to.
- CURRENT_PHASE_2: Household current on phase 2, in A. Value is very inaccurate, but can be used to identify the phase the household is connected to.
- CURRENT_PHASE_3: Household current on phase 3, in A. Value is very inaccurate, but can be used to identify the phase the household is connected to.
- EXPORT_KW: Average power output of the household in the last minute (difference of consecutive measurements of EXPORT_KWH), in kW.
- EXPORT_KWH: Total energy the household has exported since the household was connected to the smart meter, in kWh.
- GAS_USAGE_M3: Total cubic meters of gas used since it was connected to the smart meter, in m3.
- IMPORT_KW: Average power input of the household in the last minute (difference of consecutive measurements of IMPORT_KWH), in kW.
- IMPORT_KWH: Total energy the household has imported since the household was connected to the smart meter, in kWh.
- MAX_BATTERY_KW: Discharge limit of the battery, so the maximum value BATTERY_KW is allowed to attain, in kW.
- MIN_BATTERY_KW: Charge limit of the battery, so the minimum value BATTERY_KW is allowed to attain, in kW.
- MOMENTARY_EXPORT_KW: Power output from the household at the exact time of the measurement, in kW. Value is slightly inaccurate.
- MOMENTARY_IMPORT_KW: Power input from the household at the exact time of the measurement, in kW. Value is slightly inaccurate.
- MOMENTARY_PV_KW: Power output from the PV system at the exact time of the measurement, in kW. Value is slightly inaccurate.
- OPERATIONAL_STATE: Indicates what the battery did do where 0 = battery was idle, 1 = battery was charging, 2 = battery was discharging, 3 = battery had an error.
- PV_KW: Average power production of the PV system in the last minute (difference of consecutive measurements of PV_KWH), in kW.
- PV_KWH: Total energy produced by the PV system since the PV system was connected to the pulse meter, in kWh.
- REQ_CHARGE_MODE: The CHARGE_MODE that was requested from the battery.
- STATE_OF_CHARGE: The state of charge of the battery as a percentage of the capacity. Value is extremely inaccurate.
- TOTAL_KW: Average power usage of the household in the last minute (neg. means exporting power, difference of consecutive measurements of TOTAL_KWH), in kW.
- TOTAL_KWH: Total energy the household has used since the household was connected to the smart meter (neg. means exported energy, difference of EXPORT_KWH and IMPORT_KWH), in kWh.
- UNC_KW: Average power usage of the household excluding PV and battery in the last minute, in kW.

❖ "value":
- X: The value of the measurement (unit indicated in the measurement explanation).

Figure 16 depicts the energy flow on a household. The data in this dataset was collected by installing an energy management system (EMS) in each household. These EMS were connected to the P4 port on the smart meter and read out the consumption data once per minute. Furthermore, if a battery was present, the battery management system was separately connected to the EMS. Also, if a PV system was present, a pulse meter was installed and connected to the EMS to separately measure the output. All this data was then sent over Wi-Fi to a cloud. Solar panels produce energy that can be either used directly or stored in the battery. In case of redundancy the extra energy can be redirected to the grid. Battery could also perform the same action. The total consumption of the household is calculated considering the amount of energy consumed minus the amount of energy produced; $IMPORT\_KW - EXPORT\_KW$.

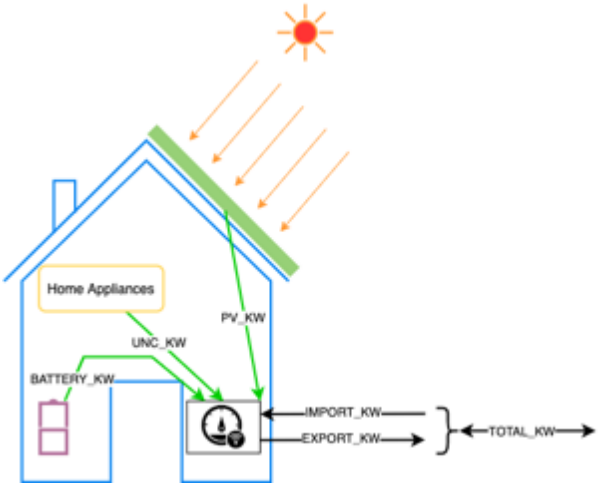

Figure 16: Energy flow in a household.

4.2.2. Weather data

We consider that the house data are in The Netherlands, hence the weather data have been retrieved from weather station 278 – Heino, which is 15km from the houses. Source: Royal Dutch Meteorological Institute (KNMI) https://www.knmi.nl/nederland-nu/klimatologie/uurgegevens.

Data attributes are described below:

- YYYYMMDD = Date (YYYY = year, MM = month, DD = day)
- HH = Time (HH hour/hour, UT. 12 UT = 13 CET, 14 MES. Hourly division 05 runs from 04.00 UT to 5.00 UT)
- DD = Mean wind direction (in degrees) during the 10-minute period preceding the time of observation (360 = north, 90 = east, 180 = south, 270 = west, 0 = calm, 990 = variable)
- FH = Hourly mean wind speed (in 0.1 m/s)
- FF = Wind speed (in 0.1 m/s) during the 10-minute period preceding the time of observation
- FX = Maximum wind gust (in 0.1 m/s) during the hourly division
- T = Temperature (in 0.1 degrees Celsius) at 1.50 m at the time of observation
- T10N = Minimum temperature (in 0.1 degrees Celsius) at 0.1 m in the preceding 6-hour period
- TD = Dew point temperature (in 0.1 degrees Celsius) at 1.50 m at the time of observation
- SQ = Sunshine duration (in 0.1 hour) during the hourly division, calculated from global radiation (-1 for <0.05 hour))
- Q = Global radiation (in J/cm2) during the hourly division
- DR = Precipitation duration (in 0.1 hour) during the hourly division
- RH = Hourly precipitation amount (in 0.1 mm) (-1 for <0.05 mm)
- P = Air pressure (in 0.1 hPa) reduced to mean sea level, at the time of observation
- VV = Horizontal visibility at the time of observation (0 = less than 100m, 1 = 100-200m, 2 = 200-300m, ..., 49 = 4900-5000m, 50 = 5-6km, 56 = 6-7km, 57 = 7-8km, ..., 79 = 29-30km, 80 = 30-35km, 81 = 35-40km, ..., 89 = more than 70km)
- N = Cloud cover (in octants), at the time of observation (9 = sky invisible)
- U = Relative atmospheric humidity (in percent) at 1.50 m at the time of observation
- WW = Present weather code (00-99), description for the hourly division.
- IX = Indicator present weather code (1 = manned and recorded (using code from visual observations), 2-3 = manned and omitted (no significant weather phenomenon to report, not available), 4 = automatically recorded (using code from visual observations), 5-6 = automatically omitted (no significant weather phenomenon to report, not available), 7 = automatically set (using code from automated observations)
- M = Fog 0 = no occurrence, 1 = occurred during the preceding hour and/or at the time of observation
- R = Rainfall 0 = no occurrence, 1 = occurred during the preceding hour and/or at the time of observation
- S = Snow 0 = no occurrence, 1 = occurred during the preceding hour and/or at the time of observation

- O = Thunder 0 = no occurrence, 1 = occurred during the preceding hour and/or at the time of observation
- Y = Ice formation 0 = no occurrence, 1 = occurred during the preceding hour and/or at the time of observation

4.3. Data Preprocessing

The main goal of data preprocessing, in our case, was to end up with a single csv file for each (77) house containing columns/features both from energy and weather data set. Energy data set entries are in a minute interval and on the other hand weather data are on an hour interval; hence energy data have also been aggregated per hour. The file retrieved from [99] has a size of 66GB.

4.3.1. Energy Consumption Data

Figure 17 shows a sample of the top 200k rows of the raw data file. Data seem out of order, however within the same house, e.g., House6, and the same measurement, e.g., BATEERY_KW, they are in chronological order.

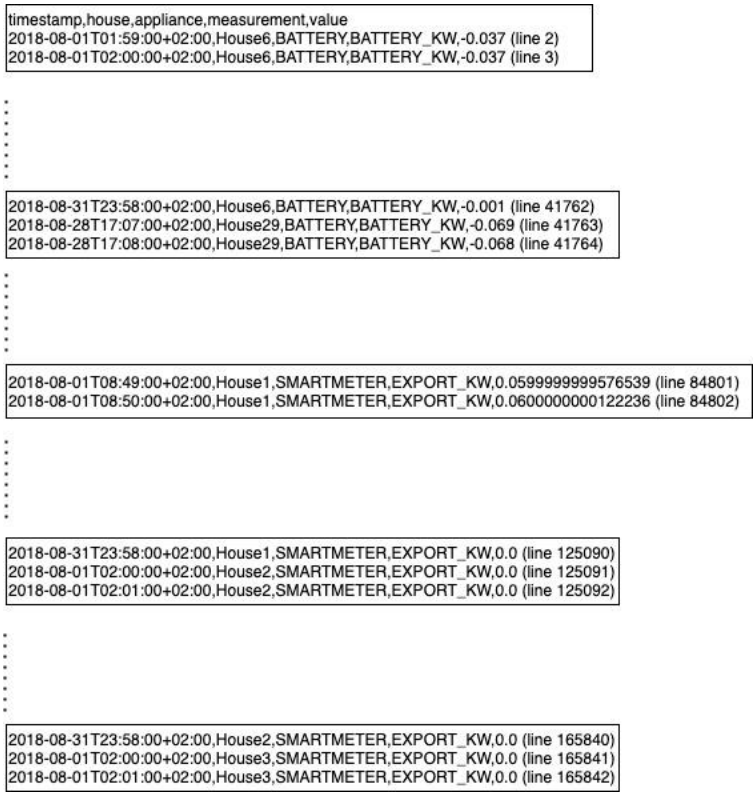

Figure 17: Raw data - sample of top 200k rows.

Carefully following the data description on the previous section 4.2.1, we notice that some data points are marked as inaccurate e.g., CURRENT_PHASE_1, some are related to gas e.g., GAS_USAGE_M3, some indicating modes e.g., REQ_CHARGE_MODE, some are totals since the connection e.g., BATTERY_IMPORT_KW. Our first decision was to limit the size of the file by keeping only the averages per minute values for consumption, production, or storage of energy. These values were: IMPORT_KW, EXPORT_KW, PV_KW, BATTERY_KW. For streaming such a large file, we used the Dask [100], a flexible open-source Python library for parallel computing. The size dropped to the reasonable amount of 11.75GB, see Figure 18.

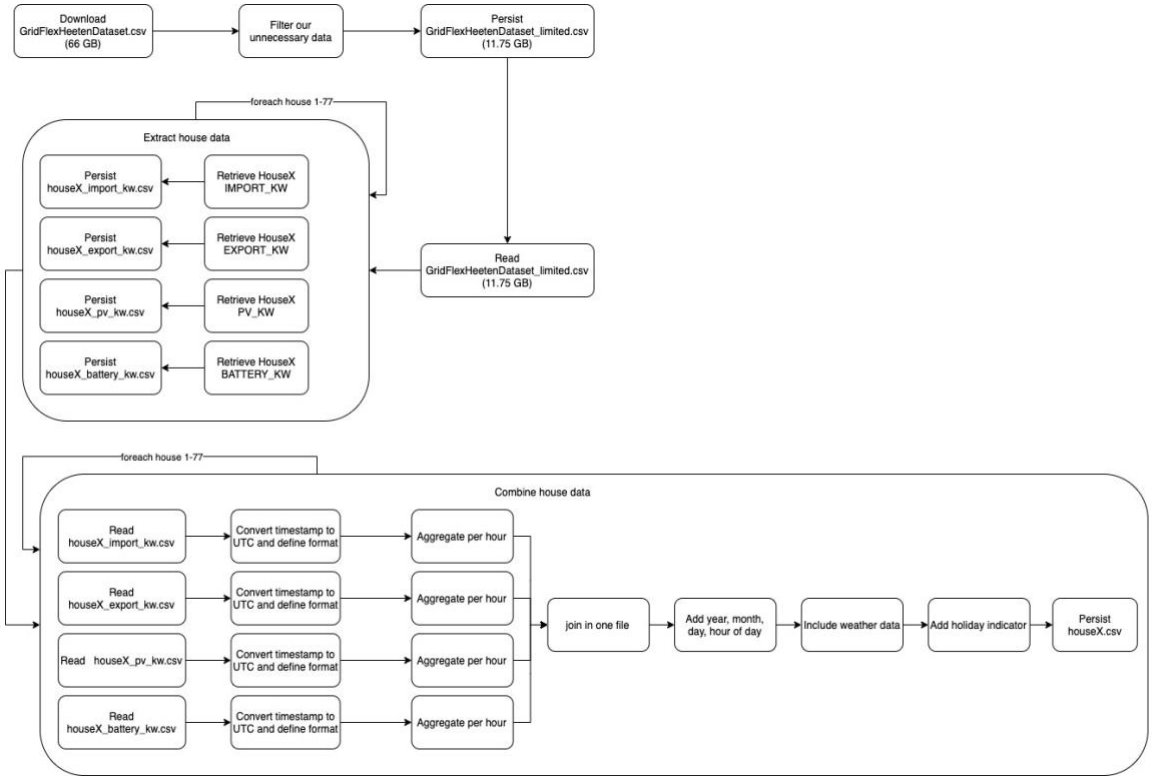

Figure 18: Energy Data - Preprocessing flow

Since data are in order within the same house and measurements, we extract the different measurement to separate csv files per house. Then for each of those files we convert the timestamp to UTC in order to get rid of the daylight savings and the change the date format to %Y-%m-%d %H:%M:%S. Trying to reduce a bit our data points and to be easy to join with weather data, each file is aggregated per hour. Thereafter, files are joined per house. Extra features like year, month, day, hour of day, holiday indicator (is weekend or public holiday), and weather data have added.

The result for house1 is shown below, Figure 19.

timestamp	IMPORT_KW	EXPORT_KW	PV_KW	BATTERY_KW	YEAR	MONTH	DAY	HOUR_OF_DAY	DD	FH	FF	FX	T	TD	SQ	Q	DR	RH	U	IX	HOLIDAY
2018-08-01 00:00:00	11.58	0.0	0.0	0.0	2018	8	1	0	0	0	0	10	145	131	0	0	0	0	91	6	0
2018-08-01 01:00:00	13.44	0.0	0.0	0.0	2018	8	1	1	100	0	10	10	125	122	0	0	0	0	97	6	0
2018-08-01 02:00:00	16.50	0.0	0.0	0.0	2018	8	1	2	0	0	0	10	127	125	0	0	0	0	98	6	0
2018-08-01 03:00:00	14.94	0.0	0.0	0.0	2018	8	1	3	0	0	0	10	124	123	0	0	0	0	99	6	0
2018-08-01 04:00:00	9.06	0.0	0.0	0.0	2018	8	1	4	230	0	10	20	116	112	0	0	0	0	97	6	0

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 18286 entries, 2018-08-01 00:00:00 to 2020-08-31 21:00:00
Data columns (total 21 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   IMPORT_KW     18286 non-null  float64
 1   EXPORT_KW     18286 non-null  float64
 2   PV_KW         18286 non-null  float64
 3   BATTERY_KW    18286 non-null  float64
 4   YEAR          18286 non-null  int64
 5   MONTH         18286 non-null  int64
 6   DAY           18286 non-null  int64
 7   HOUR_OF_DAY   18286 non-null  int64
 8   DD            18286 non-null  int64
 9   FH            18286 non-null  int64
 10  FF            18286 non-null  int64
 11  FX            18286 non-null  int64
 12  T             18286 non-null  int64
 13  TD            18286 non-null  int64
 14  SQ            18286 non-null  int64
 15  Q             18286 non-null  int64
 16  DR            18286 non-null  int64
 17  RH            18286 non-null  int64
 18  U             18286 non-null  int64
 19  IX            18286 non-null  int64
 20  HOLIDAY       18286 non-null  int64
dtypes: float64(4), int64(17)
memory usage: 3.1 MB
```

	IMPORT_KW	EXPORT_KW	PV_KW	BATTERY_KW	YEAR	MONTH	DAY	HOUR_OF_DAY	DD	FH
count	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000
mean	15.781916	17.086419	-24.369456	-0.667648	2019.119326	6.579897	15.747676	11.498797	196.761457	30.132342
std	18.196709	33.447393	41.839902	7.802271	0.711852	3.391530	8.809104	6.921796	134.717037	18.506855
min	0.000000	0.000000	-180.000000	-30.192000	2018.000000	1.000000	1.000000	0.000000	0.000000	0.000000
25%	2.520000	0.000000	-30.960000	-0.001000	2019.000000	4.000000	8.000000	5.250000	110.000000	20.000000
50%	11.400000	0.000000	-0.180000	0.000000	2019.000000	7.000000	16.000000	11.000000	210.000000	30.000000
75%	23.100000	14.940000	0.000000	0.000000	2020.000000	9.000000	23.000000	17.000000	250.000000	40.000000
max	242.940000	165.420000	0.000000	30.018000	2020.000000	12.000000	31.000000	23.000000	990.000000	130.000000

	FF	FX	T	TD	SQ	Q	DR	RH	U	IX	HOLIDAY
count	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.0	18286.000000
mean	30.261949	59.785628	112.603030	75.164826	2.262715	46.541343	0.651482	0.755223	79.625123	6.0	0.307120
std	18.951812	33.944219	68.525315	55.123949	3.680945	73.909376	2.146047	4.335331	16.511974	0.0	0.461312
min	0.000000	0.000000	-84.000000	-91.000000	0.000000	0.000000	0.000000	-1.000000	18.000000	6.0	0.000000
25%	20.000000	30.000000	62.000000	32.000000	0.000000	0.000000	0.000000	0.000000	69.000000	6.0	0.000000
50%	30.000000	50.000000	106.000000	74.000000	0.000000	2.000000	0.000000	0.000000	84.000000	6.0	0.000000
75%	40.000000	80.000000	161.000000	119.000000	4.000000	69.000000	0.000000	0.000000	93.000000	6.0	1.000000
max	140.000000	270.000000	387.000000	241.000000	10.000000	329.000000	10.000000	150.000000	100.000000	6.0	1.000000

Figure 19: House 1 data, first 5 rows (top), column info (middle), stats (bottom).

4.3.2. Weather data

Figure 20 shows the format of the raw data (as described in 4.2.2), the types of the columns and some stats regarding the data.

	STN	YYYYMMDD	HH	DD	FH	FF	FX	T	T10N	TD	SQ	Q	DR	RH	P	VV	N	U	WW	IX	M	R	S	O	Y
0	278	20180801	1	100	0	10	10	125	NaN	122	0	0	0	0	NaN	NaN	NaN	97	NaN	6	NaN	NaN	NaN	NaN	NaN
1	278	20180801	2	0	0	0	10	127	NaN	125	0	0	0	0	NaN	NaN	NaN	98	NaN	6	NaN	NaN	NaN	NaN	NaN
2	278	20180801	3	0	0	0	10	124	NaN	123	0	0	0	0	NaN	NaN	NaN	99	NaN	6	NaN	NaN	NaN	NaN	NaN
3	278	20180801	4	230	0	10	20	116	NaN	112	0	0	0	0	NaN	NaN	NaN	97	NaN	6	NaN	NaN	NaN	NaN	NaN
4	278	20180801	5	80	0	10	10	133	NaN	131	5	15	0	0	NaN	NaN	NaN	98	NaN	6	NaN	NaN	NaN	NaN	NaN

	STN	DD	FH	FF	FX	T	T10N	TD	SQ	Q
count	18288.0	18288.000000	18288.00000	18288.000000	18288.000000	18288.00000	3048.000000	18288.000000	18288.000000	18288.000000
mean	278.0	196.768920	30.13014	30.259733	59.781277	112.60362	79.538714	75.167870	2.262467	46.536253
std	0.0	134.711651	18.50704	18.951961	33.944912	68.52162	68.776330	55.121703	3.680819	73.906936
min	278.0	0.000000	0.00000	0.000000	0.000000	-84.00000	-115.000000	-91.000000	0.000000	0.000000
25%	278.0	110.000000	20.00000	20.000000	30.000000	62.00000	29.000000	32.000000	0.000000	0.000000
50%	278.0	210.000000	30.00000	30.000000	50.000000	106.00000	74.500000	74.000000	0.000000	2.000000
75%	278.0	250.000000	40.00000	40.000000	80.000000	161.00000	130.000000	119.000000	4.000000	69.000000
max	278.0	990.000000	130.00000	140.000000	270.000000	387.00000	322.000000	241.000000	10.000000	329.000000

	DR	RH	P	VV	N	U	WW	IX	M	R	S	O	Y
count	18288.000000	18288.000000	0.0	0.0	0.0	18288.000000	0.0	18288.0	0.0	0.0	0.0	0.0	0.0
mean	0.651411	0.755140	NaN	NaN	NaN	79.626258	NaN	6.0	NaN	NaN	NaN	NaN	NaN
std	2.145940	4.335102	NaN	NaN	NaN	16.511481	NaN	0.0	NaN	NaN	NaN	NaN	NaN
min	0.000000	-1.000000	NaN	NaN	NaN	18.000000	NaN	6.0	NaN	NaN	NaN	NaN	NaN
25%	0.000000	0.000000	NaN	NaN	NaN	69.000000	NaN	6.0	NaN	NaN	NaN	NaN	NaN
50%	0.000000	0.000000	NaN	NaN	NaN	84.000000	NaN	6.0	NaN	NaN	NaN	NaN	NaN
75%	0.000000	0.000000	NaN	NaN	NaN	93.000000	NaN	6.0	NaN	NaN	NaN	NaN	NaN
max	10.000000	150.000000	NaN	NaN	NaN	100.000000	NaN	6.0	NaN	NaN	NaN	NaN	NaN

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 18288 entries, 0 to 18287
Data columns (total 25 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   STN       18288 non-null  int64
 1   YYYYMMDD  18288 non-null  object
 2   HH        18288 non-null  object
 3   DD        18288 non-null  int64
 4   FH        18288 non-null  int64
 5   FF        18288 non-null  int64
 6   FX        18288 non-null  int64
 7   T         18288 non-null  int64
 8   T10N      3048 non-null   float64
 9   TD        18288 non-null  int64
 10  SQ        18288 non-null  int64
 11  Q         18288 non-null  int64
 12  DR        18288 non-null  int64
 13  RH        18288 non-null  int64
 14  P         0 non-null      float64
 15  VV        0 non-null      float64
 16  N         0 non-null      float64
 17  U         18288 non-null  int64
 18  WW        0 non-null      float64
 19  IX        18288 non-null  int64
 20  M         0 non-null      float64
 21  R         0 non-null      float64
 22  S         0 non-null      float64
 23  O         0 non-null      float64
 24  Y         0 non-null      float64
dtypes: float64(10), int64(13), object(2)
memory usage: 3.6+ MB
```

Figure 20: Top 5 lines of weather data set (top), column types (bottom right), stats (bottom left)

Initially, empty columns (P, VV, N, WW, M, R, S, O, Y) have been dropped. Column T10N was present only for 3048 entries; hence we drop it as well. STN was the last column to be dropped since it is the weather station id, which in our case is irrelevant. Next task was to create a timestamp column, by combining columns YYYYMMD and HH with format '%Y%m%d%H'. Then columns YYYYMMD and HH have been dropped. The final output is shown in Figure 21.

timestamp	DD	FH	FF	FX	T	TD	SQ	Q	DR	RH	U	IX
2018-08-01 01:00:00	100	0	10	10	125	122	0	0	0	0	97	6
2018-08-01 02:00:00	0	0	0	10	127	125	0	0	0	0	98	6
2018-08-01 03:00:00	0	0	0	10	124	123	0	0	0	0	99	6
2018-08-01 04:00:00	230	0	10	20	116	112	0	0	0	0	97	6
2018-08-01 05:00:00	80	0	10	10	133	131	5	15	0	0	98	6

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 18288 entries, 2018-08-01 01:00:00 to 2020-08-31 00:00:00
Data columns (total 12 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   DD      18288 non-null  int64
 1   FH      18288 non-null  int64
 2   FF      18288 non-null  int64
 3   FX      18288 non-null  int64
 4   T       18288 non-null  int64
 5   TD      18288 non-null  int64
 6   SQ      18288 non-null  int64
 7   Q       18288 non-null  int64
 8   DR      18288 non-null  int64
 9   RH      18288 non-null  int64
 10  U       18288 non-null  int64
 11  IX      18288 non-null  int64
dtypes: int64(12)
memory usage: 1.8 MB
```

	DD	FH	FF	FX	T	TD	SQ	Q	DR	RH	U	IX
count	18288.000000	18288.00000	18288.000000	18288.000000	18288.00000	18288.000000	18288.000000	18288.000000	18288.000000	18288.000000	18288.000000	18288.0
mean	196.768920	30.13014	30.259733	59.781277	112.60362	75.167870	2.262467	46.536253	0.651411	0.755140	79.626258	6.0
std	134.711651	18.50704	18.951961	33.944912	68.52162	55.121703	3.680819	73.906936	2.145940	4.335102	16.511481	0.0
min	0.000000	0.00000	0.000000	0.000000	-84.00000	-91.000000	0.000000	0.000000	0.000000	-1.000000	18.000000	6.0
25%	110.000000	20.00000	20.000000	30.000000	62.00000	32.000000	0.000000	0.000000	0.000000	0.000000	69.000000	6.0
50%	210.000000	30.00000	30.000000	50.000000	106.00000	74.000000	0.000000	2.000000	0.000000	0.000000	84.000000	6.0
75%	250.000000	40.00000	40.000000	80.000000	161.00000	119.000000	4.000000	69.000000	0.000000	0.000000	93.000000	6.0
max	990.000000	130.00000	140.000000	270.000000	387.00000	241.000000	10.000000	329.000000	10.000000	150.000000	100.000000	6.0

Figure 21: Weather data cleaned and formatted.

4.4. Exploratory Data Analysis (EDA)

Exploratory Data Analysis refers to the critical process of performing an initial investigation on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. After pre-processing, we ended up with 77 csv files, one for each house. The first step was to visualize the consumption data of all houses to identify patterns, trends, missing values.

4.4.1. Data Visualization after pre-processing

Data visualization is one of the steps of the data science process, which states that after data has been collected, processed and modeled, it must be visualized for conclusions to be made.Visualization is an increasingly key tool to make sense of the trillions of rows of data generated every day. On top it is useful for data cleaning, exploring data structure, detecting outliers and unusual groups, identifying trends and clusters, spotting local patterns, evaluating modeling output, and presenting results. It is essential for exploratory data analysis and data mining to check data quality and to help analysts become familiar with the structure and features of the data before them. Data visualization helps to tell stories by curating data into a form easier to understand, highlighting the trends and outliers. A good visualization tells a story, removing the noise from data and highlighting useful information. Graphics reveal data features that statistics and models may miss: unusual distributions of data, local patterns, clusterings, gaps, missing values, evidence of rounding or heaping, implicit boundaries, outliers, and so on. Graphics raise questions that stimulate research and suggest ideas. In fact, interpreting graphics needs experience to identify potentially interesting features and statistical nous to guard against the dangers of overinterpretation. Just as graphics are useful for checking model results, models are useful for checking ideas derived from graphics.

Since the number of houses is relatively high and it wouldn’t be intuitive to display all of them, we took as reference three houses: house 2, house 3, house 69. The main reason was that we identified different consumption patterns; thus, was interesting to dive in further. In this work we will focus only on consumption data IMPORT_KW column in combination with weather data. Features like EXPORT_KW, PV_KW and BATTERY_KW won’t be considered since their data are incomplete and we need more business context on how to use them.

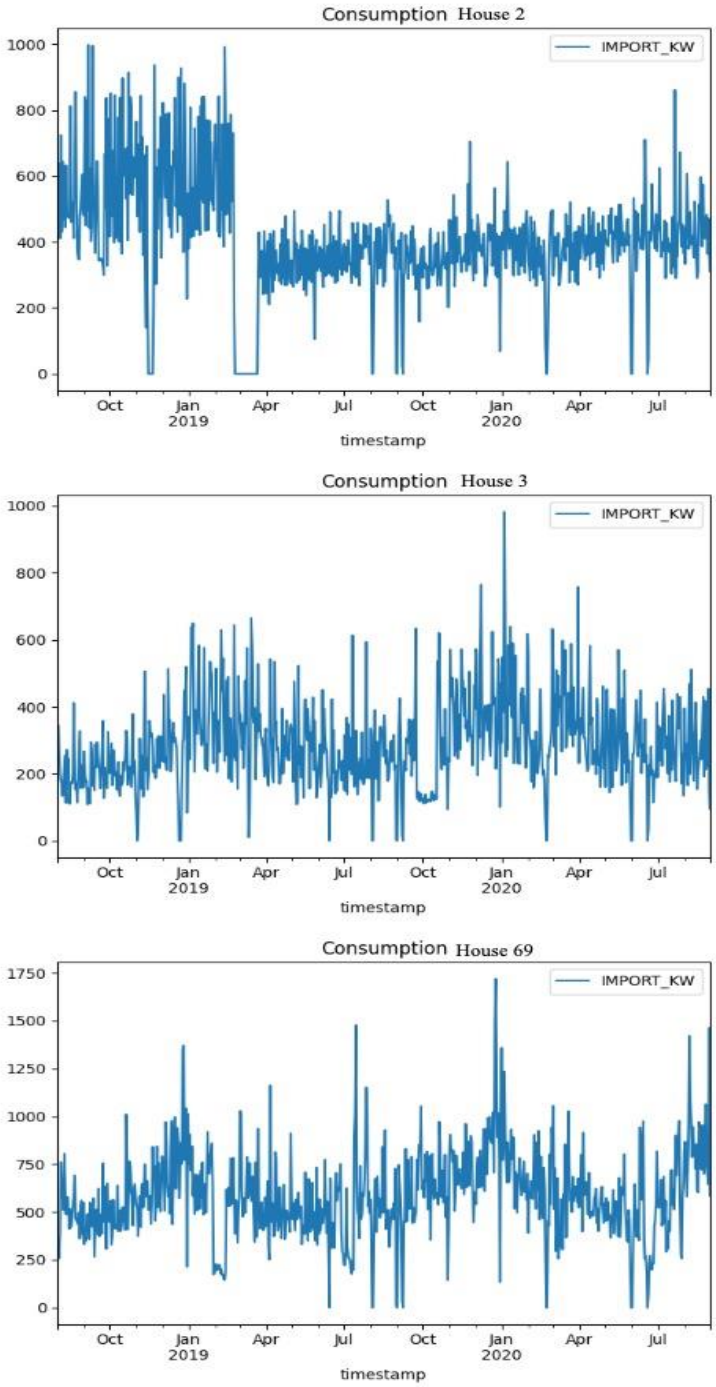

Figure 22: House 2 (top), House 3 (middle), House 69 (bottom). Data after preprocessing.

Houses 3 and 69 show a clear seasonal pattern, where the consumption during winter is higher that summer. On the opposite side this does not seem to be the case in house 2. House 2 has higher consumption from August 2018 – February 2019, then there is period where data are missing and then the pattern is almost the same for the rest months. House 69 seems to have higher average consumption that the other two houses. In all three houses there are steep valleys, in such cases the smart meter didn't work for a small period; hours to a few days, resulting in missing values. Steep peaks are much more challenging to interpret. These might be related to erroneous values reported from the meter.

Figure 23 below visualizes the weather data set. As expected, weather attributes e.g., Temperature, follow a seasonal pattern.

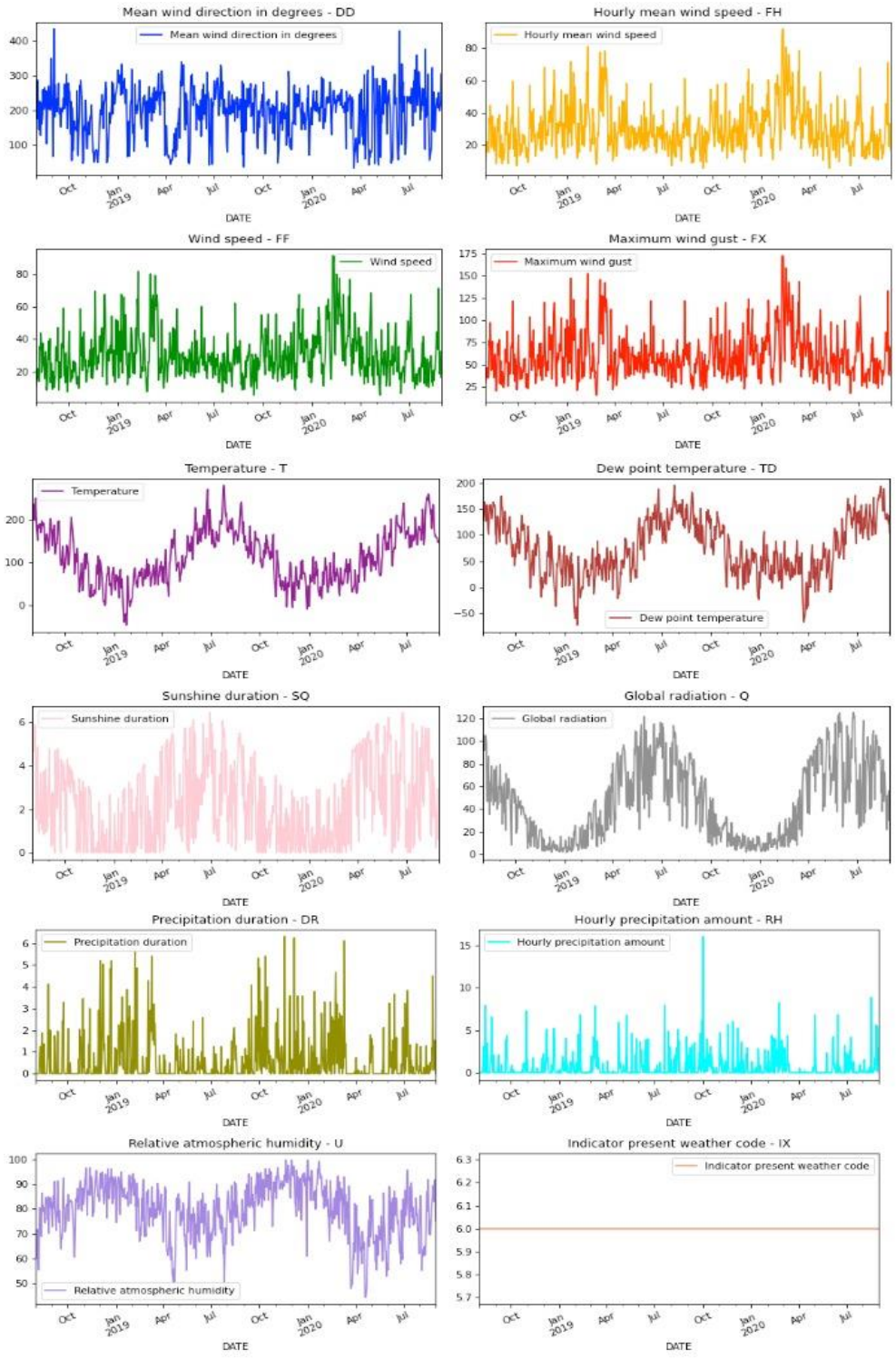

Figure 23: Weather data.

4.4.2. Data Cleaning

After going through the plots of all houses, we concluded that is necessary to proceed with filtering out houses with erratic data, replacing missing values and smooth out steep peaks that were hard to explain.

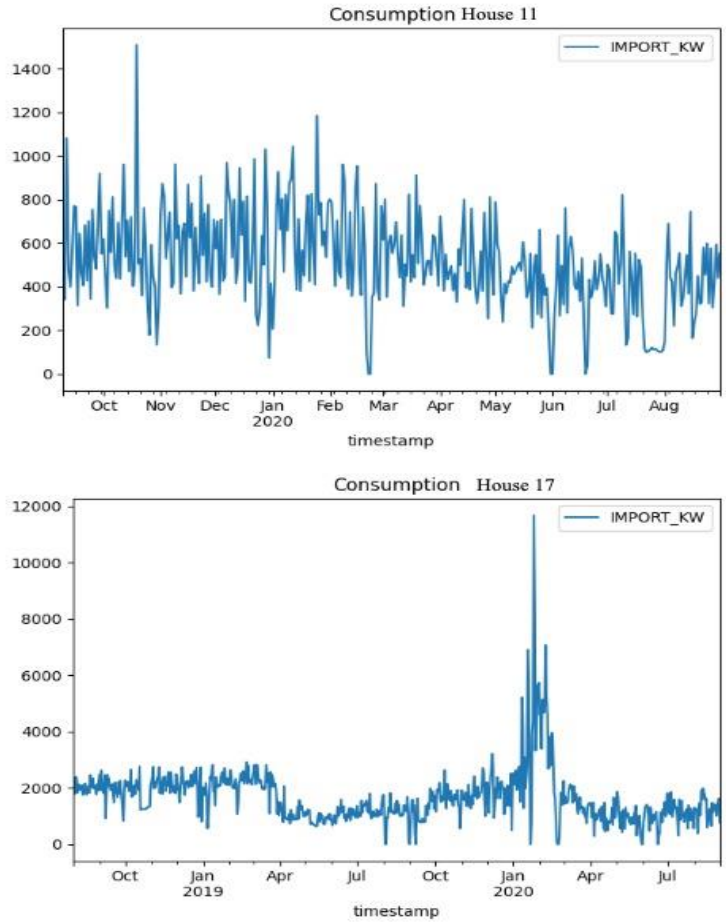

Figure 24: Example of houses with erratic data. House 11 (top), House 17 (bottom).

Figure 24 above, depicts two houses with erratic data. Let's first define erratic in our case. Houses with erratic data could be either houses with incomplete data and more specifically not a complete year; see house 11, or houses that do not have a consumption pattern; see house 17. The following houses, eighteen houses in total, have been filtered out: 11, 12, 13, 16, 17, 34, 36, 38, 40, 42, 49, 58, 59, 65, 67, 68, 71, 74, 76.

For the remaining fifty seven houses, we identified the missing values and the results are shown in the following table.

Table 3: Overview of missing values count.

House id	Missing values count	Total rows
1	2601	18286
2	1396	18286
3	1097	18286
4	3598	18286
5	3206	18286
6	4869	18274
7	1451	18286
8	529	18286
9	2268	18286
10	3765	18286
14	2952	18286
15	1896	18286
18	1441	16402
19	3223	18286
20	2731	18286
21	4067	18195

22	534	18286
23	1773	18286
24	2678	18286
25	3263	18286
26	379	15993
27	1873	18286
28	3068	18286
29	1267	18286
30	491	18286
31	4146	18286
32	4311	18002
33	2750	18286
35	4110	18020
37	570	18286
39	3494	18286
41	1851	18286
43	2054	18231
44	1086	18286
45	2045	18286
46	2736	18286
47	2031	18286
48	1336	18286
50	1346	18286
51	1664	18286
52	575	18286
53	4402	18286
54	489	18286
55	719	18286
56	2782	18286
57	3809	18286
60	3919	18286
61	498	18286
62	714	18286
63	1957	18286
64	3043	18286
66	1821	18286
69	1209	18286
70	2668	18286
72	683	18286
73	517	18286
75	2498	18286
77	3084	18286

There are some entries marked in orange. These houses miss some entries either in the beginning and/or the end of the full period August 2018 – August 2020, so when replacing missing values, we should also consider those periods. Missing data replacement mechanism considers two cases:

- Long periods > week.
- Short periods <= week.

For the first case, we replace the missing values from the respective period either in previous or next year; depending on the availability of the data. Noise was introduced so the values are not exactly the same. This approach was based on the fact that electricity consumption, Figure 22, and weather, Figure 23, show a seasonal pattern. For short periods, the values have been replaced from the median of the month adding noise on top.

As mentioned before house data contain some high peaks that are not easy to interpret. We define high peak as a value three (rule of thumb) times higher than the median of the month. We replace high peaks with the median of the month introducing again noise.

Table below shows house statistics after data cleaning process.

Table 4: House data statistics after replacing missing data.

House id	Min kW	Max kW	Mean kW	Median kW
1	0.06	34.14	12.88	11.32
2	0.12	40.08	14.93	13.15
3	0.06	24.30	8.23	7.92
4	0.06	27.13	8.88	7.83
5	0.002	27.54	9.58	8.88
6	0.001	76.93	26.34	25.04
7	0.06	32.52	10.36	10.02
8	0.12	55.80	20.95	17.58
9	0.60	30.96	11.37	9.91
10	0.003	35.99	12.69	11.46
14	0.06	20.52	6.97	6.71
15	0.06	23.22	7.87	7.74
18	0.004	14.27	4.34	4.32
19	0.01	27.66	9.16	8.76
20	0.06	20.28	7.56	6.71
21	0.003	98.23	33.89	31.35
23	0.06	88.62	31.73	29.71
24	0.06	40.08	13.95	13.35
25	0.06	36.72	12.92	11.94
26	0.01	36.55	10.51	9.34
27	0.09	67.14	23.13	21.64
28	0.06	59.76	21.59	19.07
29	0.06	75.78	24.49	23.84
30	0.24	52.67	20.28	17.03
31	0.0008	21.42	6.98	7.02
32	0.002	37.44	13.35	12.24
33	0.02	24.66	9.03	8.37
35	0.82	28.62	10.33	9.57
37	0.24	48.18	18.14	15.60
39	0.03	45.72	16.45	14.16
41	0.24	72.84	27.66	24.18
43	0.06	114.66	41.69	37.93
44	0.06	61.73	24.13	20.10
45	0.02	33.84	12.37	10.98
46	0.02	19.07	6.61	5.84
47	0.29	72.01	26.83	23.58
48	0.06	35.94	12.80	11.27
50	0.06	23.51	7.82	7.84
51	0.06	43.74	15.37	14.82
52	0.06	17.81	6.62	5.52
53	0.44	62.01	25.46	22.56
54	0.11	38.46	14.98	12.50
55	0.06	53.64	19.85	17.28
56	0.06	17.82	6.23	5.34
57	0.23	37.97	14.51	12.84
60	0.06	29.45	10.73	9.72
61	0.18	57.24	20.91	18.34
62	0.12	31.31	12.03	9.96
63	0.06	38.34	13.32	12.23
64	0.06	36.54	12.64	11.54
66	0.008	39.54	14.72	12.89
69	0.06	53.98	20.43	17.82
70	0.0005	39.60	13.15	12.35
72	0.06	45.84	15.56	13.91

73	0.11	19.74	7.46	6.59
75	0.06	30.59	10.15	9.06
77	0.06	37.74	14.58	12.16

4.4.3. Time series analysis

The analysis will be performed in house3, and similarly was applied on all houses. Let's have a look on house3 data, Figure 25.

timestamp	IMPORT_KW	YEAR	MONTH	DAY	HOUR_OF_DAY	DD	FH	FF	FX	T	TD	SQ	Q	DR	RH	U	IX	HOLIDAY
2018-08-01 00:00:00	12.24	2018	8	1	0	0	0	0	10	145	131	0	0	0	0	91	6	0
2018-08-01 01:00:00	9.36	2018	8	1	1	100	0	10	10	125	122	0	0	0	0	97	6	0
2018-08-01 02:00:00	12.06	2018	8	1	2	0	0	0	10	127	125	0	0	0	0	98	6	0
2018-08-01 03:00:00	11.64	2018	8	1	3	0	0	0	10	124	123	0	0	0	0	99	6	0
2018-08-01 04:00:00	11.76	2018	8	1	4	230	0	10	20	116	112	0	0	0	0	97	6	0

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 18286 entries, 2018-08-01 00:00:00 to 2020-08-31 21:00:00
Data columns (total 18 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   IMPORT_KW    18286 non-null  float64
 1   YEAR         18286 non-null  int64
 2   MONTH        18286 non-null  int64
 3   DAY          18286 non-null  int64
 4   HOUR_OF_DAY  18286 non-null  int64
 5   DD           18286 non-null  int64
 6   FH           18286 non-null  int64
 7   FF           18286 non-null  int64
 8   FX           18286 non-null  int64
 9   T            18286 non-null  int64
 10  TD           18286 non-null  int64
 11  SQ           18286 non-null  int64
 12  Q            18286 non-null  int64
 13  DR           18286 non-null  int64
 14  RH           18286 non-null  int64
 15  U            18286 non-null  int64
 16  IX           18286 non-null  int64
 17  HOLIDAY      18286 non-null  int64
dtypes: float64(1), int64(17)
memory usage: 2.7 MB
```

	IMPORT_KW	YEAR	MONTH	DAY	HOUR_OF_DAY	DD	FH	FF	FX
count	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000
mean	8.232439	2019.119326	6.579897	15.747676	11.498797	196.761457	30.132342	30.261949	59.785628
std	4.622721	0.711852	3.391530	8.809104	6.921796	134.717037	18.506855	18.951812	33.944219
min	0.060000	2018.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	5.700000	2019.000000	4.000000	8.000000	5.250000	110.000000	20.000000	20.000000	30.000000
50%	7.920000	2019.000000	7.000000	16.000000	11.000000	210.000000	30.000000	30.000000	50.000000
75%	9.772830	2020.000000	9.000000	23.000000	17.000000	250.000000	40.000000	40.000000	80.000000
max	24.300000	2020.000000	12.000000	31.000000	23.000000	990.000000	130.000000	140.000000	270.000000

	T	TD	SQ	Q	DR	RH	U	IX	HOLIDAY
count	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.000000	18286.0	18286.000000
mean	112.603030	75.164826	2.262715	46.541343	0.651482	0.755223	79.625123	6.0	0.307120
std	68.525315	55.123949	3.680945	73.909376	2.146047	4.335331	16.511974	0.0	0.461312
min	-84.000000	-91.000000	0.000000	0.000000	0.000000	-1.000000	18.000000	6.0	0.000000
25%	62.000000	32.000000	0.000000	0.000000	0.000000	0.000000	69.000000	6.0	0.000000
50%	106.000000	74.000000	0.000000	2.000000	0.000000	0.000000	84.000000	6.0	0.000000
75%	161.000000	119.000000	4.000000	69.000000	0.000000	0.000000	93.000000	6.0	1.000000
max	387.000000	241.000000	10.000000	329.000000	10.000000	150.000000	100.000000	6.0	1.000000

Figure 25: House 3 data, first 5 rows (top), column info (middle), stats (bottom).

At the first glance there isn't an extreme deviation between mean and median values of IMPORT_KW. As expected after data cleaning, there are no missing values, middle section of the figure above. When checking the stats at the bottom we noticed that IX weather feature is a constant with value 6. It does not bring any value to consider it, so we dropped it.

Figure 26 below depicts the distribution of house3. Data distribution is an important aspect to consider when modeling the data that will be fed in a model.

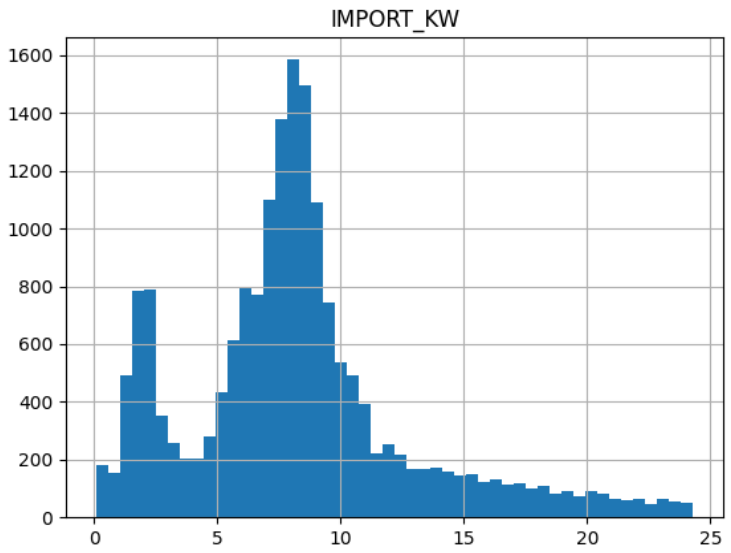

Figure 26: House 3 IMPORT_KW distribution.

Next step is to check the correlation of the features within our dataset. We notice that HOUR_OF_DAY, U, DR have a strong positive correlation with IMPORT_KW, while on the other hand T, TD, SQ, SQ, Q have a strong negative correlation. The rest of the feature do not seem to affect IMPORT_KW that much; hence we won't use them in our model.

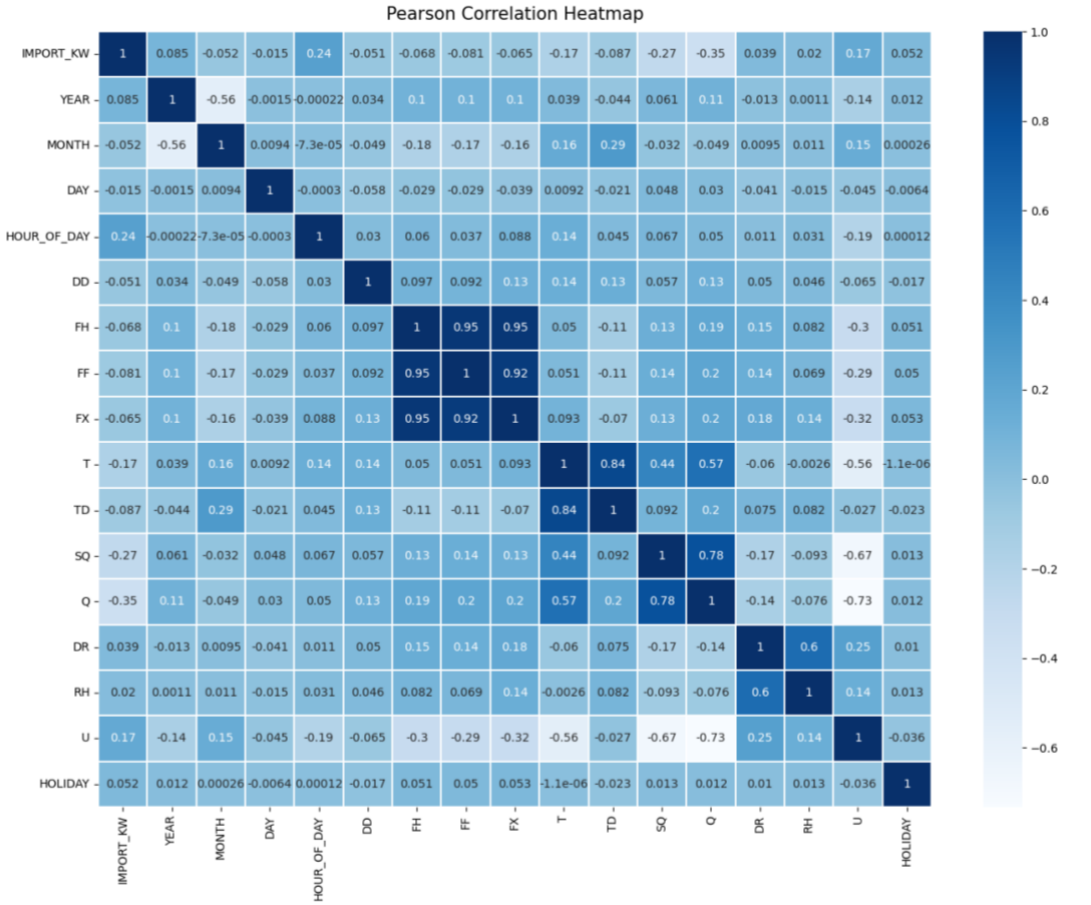

Figure 27: Pearson correlation heatmap.

Plotting the timeseries help unveiling temporal pattern. Let's take a look on house3 data. It is clear that there is a seasonal pattern. Consumption is higher during winter and lower during summer. October 2019 shows a sudden low consumption. If this is not a pattern also in the future it might bring a challenge forecasting October.

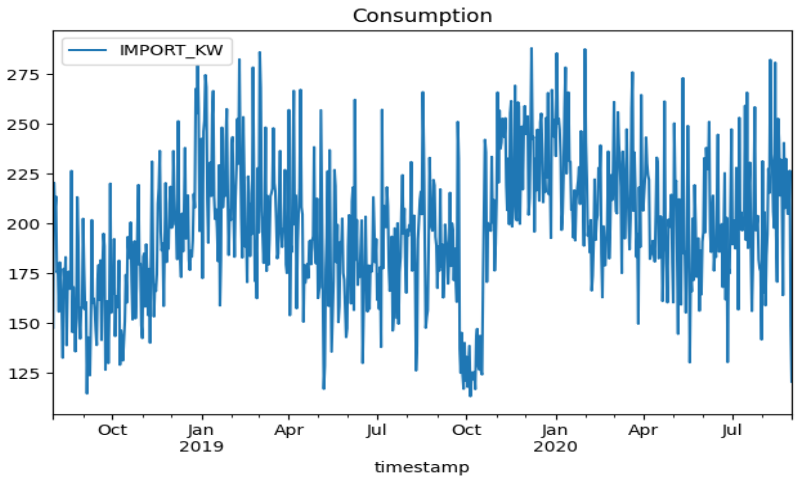

Figure 28: House 3 IMPORT_KW data

In 3.1.1, we talked about timeseries components, decomposing a timeseries helps interpreting those. We notice that the trend is following a seasonal pattern, increasing during winter, and decreasing during summer. The seasonal component also shows a pattern but we need to zoom in a bit in order to identify yearly, monthly, daily patterns. Residuals look quite steady throughout.

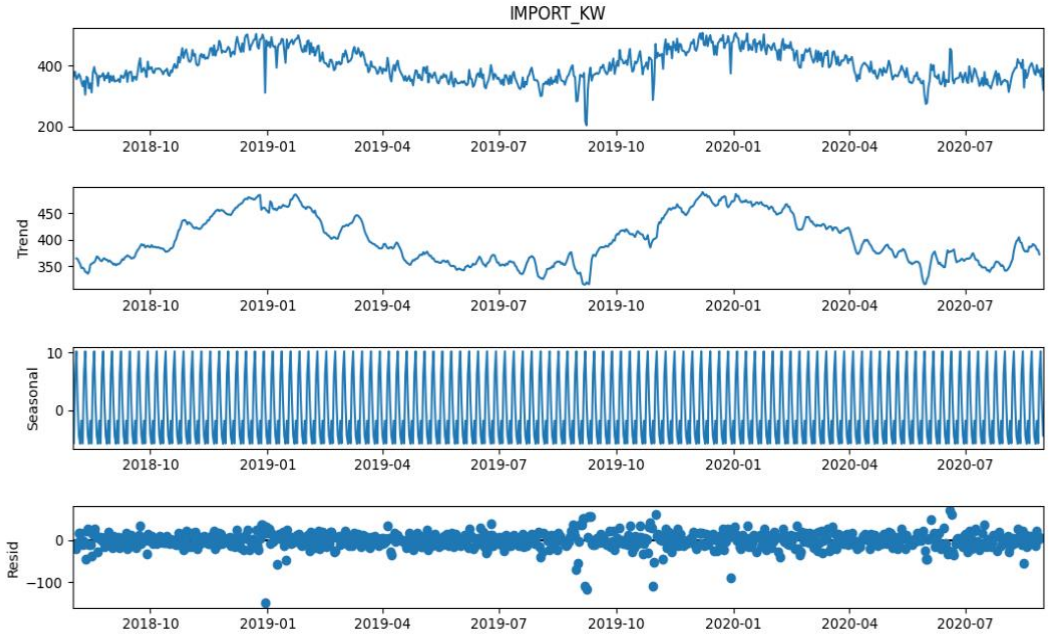

Figure 29: House 3 timeseries decomposition.

Next, we will examine the distribution of energy consumption using different internals. Boxplots are ideal for this. Boxplots are a standardized way of displaying the distribution of data based on a five number summary: minimum, first quartile Q1, median, third quartile Q3, and maximum.

- **Median (Q2/50th percentile)**: The middle value of the data set.
- **First Quartile (Q1/25th percentile)**: The median value between the smallest number of the dataset and the median of the data set.
- **Third Quartile (Q3/75th percentile)**: The median value between the median and the highest number of the dataset.
- **Interquartile Range (IQR)**: 25th to the 75th percentile.

- **maximum**: $Q3 + 1.5 * IQR$.
- **minimum**: $Q1 - 1.5 * IQR$.
- **Whiskers**: The lines which connect the minimum to Q1 and Q3 to maximum.
- **Outliers**: Values that are outside minimum or maximum.

Figure 30 depicts consumption distribution by month. The boxplot below cancels our assumption that there is a yearly pattern for this house. We see that winter months have more or less the same median values as summer ones. Low consumption occurs in May, September, and October. We also notice the presence of outliers. This plot backs up the correlation matrix Figure 27, indicating that YEAR and MONTH could be excluded as features since they do not have a considerable effect on consumption.

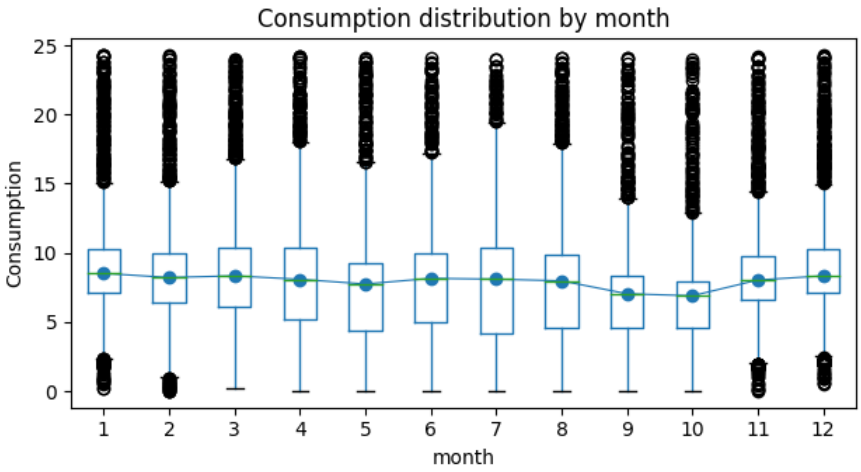

Figure 30: Distribution of energy consumption by month.

Figure 31 presents consumption distribution by week. House 3 does not show any clear weekly pattern. Outliers are present here as well. Hence, excluding feature DAY, was a decision in the right direction.

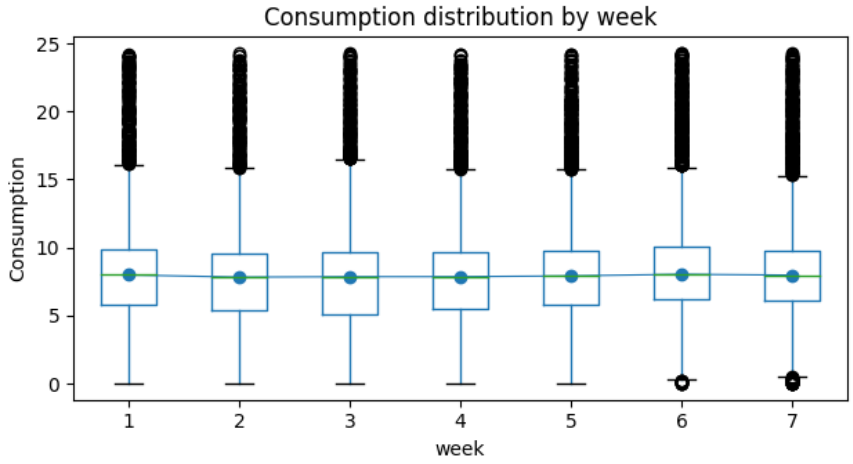

Figure 31: Distribution of energy consumption by week

It is also interesting to examine consumption distribution by hour. As shown in Figure 32 demand is relatively stable during the night. In the morning and noon, there is a drop in consumption. Assuming that heating demands during the day are less and that people are working, this pattern seems logical. From 5pm to 11pm the consumption increases. People are returning home and use the house in full capacity; heating, lighting, cooking, charging devices, charging a car, etc.
It is obvious that HOUR_OF_DAY is an important feature since it affects much the consumption.

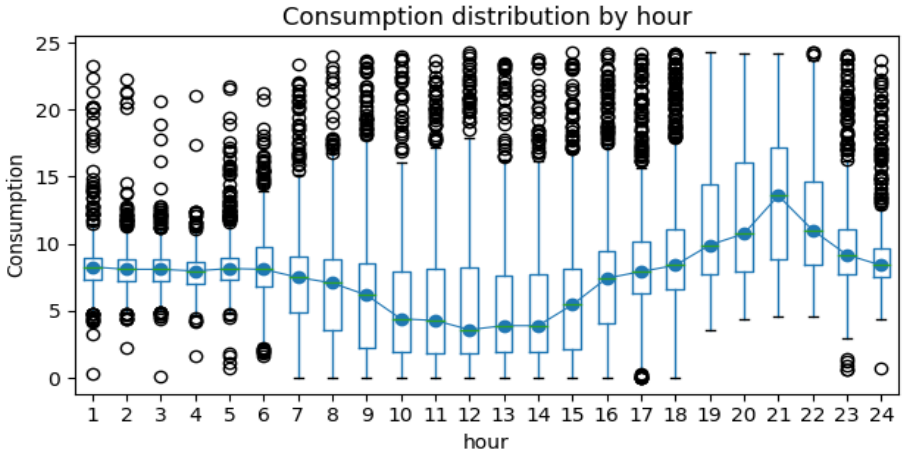

Figure 32: Distribution of energy consumption by hour.

Another interesting aspect to investigate is how consumption behaves between holidays and non-holidays, Figure 33. In our case, holidays do not have an effect on consumption; hence HOLIDAYS feature can be dropped.

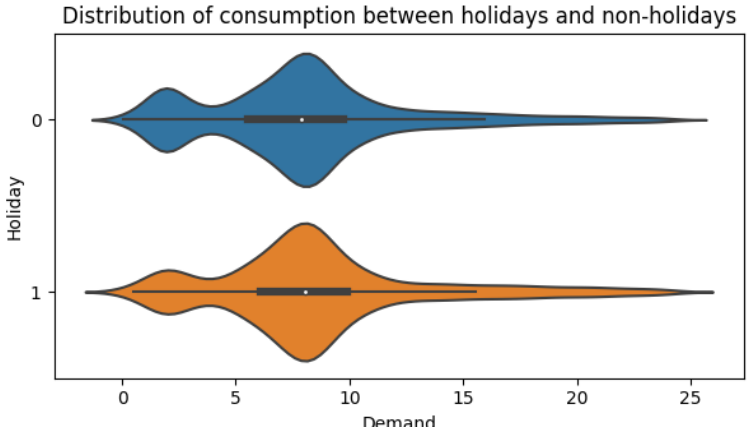

Figure 33: Distribution of consumption between holidays and non-holidays.

4.4.4. Stationary analysis

A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time. Most statistical forecasting methods assume that the time series can be rendered approximately stationary using mathematical transformations. A stationarised series is relatively easy to predict; its statistical properties will be the same in the future as they have been in the past. Another reason for trying to make the time series stationary is to be able to obtain meaningful sample statistics such as means, variances, and correlations with other variables. Such statistics are useful as descriptors of future behavior only if the series is stationary. For example, if the series is consistently increasing over time, the sample mean and variance will grow with the size of the sample, and they will always underestimate the mean and variance in future periods. And if the mean and variance of a series are not well-defined, then neither are its correlations with other variables.

Fluctuating rolling mean and standard deviation can be a first indication of non-stationary time series. Judging from Figure 34 the series does not look stationary, since the mean and the variance of timeseries are not constant over time.

Figure 34: Stationary check using rolling mean and std.

With this in mind lets perform two statistical tests to discover if series have unit root, Augmented Dickey-Fuller (ADF), or if it is trend stationary, Kwiatkowski–Phillips–Schmidt–Shin (KPSS).
ADF test:

- The null hypothesis for this test is that there is a unit root.
- The alternate hypothesis is that there is no unit root in the series.

KPSS test:

- The null hypothesis for the test is that the data is stationary.
- The alternate hypothesis for the test is that the data is not stationary.

If we fail to reject the null hypothesis, we can say that the series is non-stationary. If both mean and standard deviation are flat lines (constant mean and constant variance), the series becomes stationary.

Figure 35, shows the results of both tests, applying differencing only once is enough to make our time series is stationary. This will be our parameter d = 1 for the SARIMA model.

d	adf_stats	p-value	is_adf_stationary	is_kpss_stationary	is_stationary
0	8.416788e-24	0.01	True	False	False
1	0.000000e+00	0.10	True	True	True
2	0.000000e+00	0.10	True	True	True

Figure 35: Results of ADF and KPSS tests.

4.4.5. Autocorrelation and partial autocorrelation analysis

Autocorrelation Function (ACF) is a statistical correlation, which summarizes the strength of the relationship between two variables. Pearson's correlation coefficient is a number between -1 and 1 that describes a negative or positive correlation respectively. A value of zero indicates no correlation. We can calculate the correlation for time series observations with previous time steps, called lags. Because the correlation of the time series observations is calculated with values of the same series at previous times, this is called a serial correlation, or an autocorrelation. A plot of the autocorrelation of a time series by lag is called the Autocorrelation Function, or the acronym ACF. This plot is sometimes called a correlogram or an autocorrelation plot.

Partial Autocorrelation Function (PACF) is a summary of the relationship between an observation in a time series with observations at prior time steps with the relationships of intervening observations removed. The partial autocorrelation at lag k is the correlation that results after removing the effect of any correlations due to the terms at shorter lags. The autocorrelation for observation and observation at a prior time step is comprised of both the

direct correlation and indirect correlations. It is these indirect correlations that the partial autocorrelation function seeks to remove.

The autocorrelation and partial autocorrelation plots show a clear association between one hour's consumption and previous hours, as well as between one hour's consumption and the same hour's consumption on previous days. This type of correlation is an indication that autoregressive models can work well.

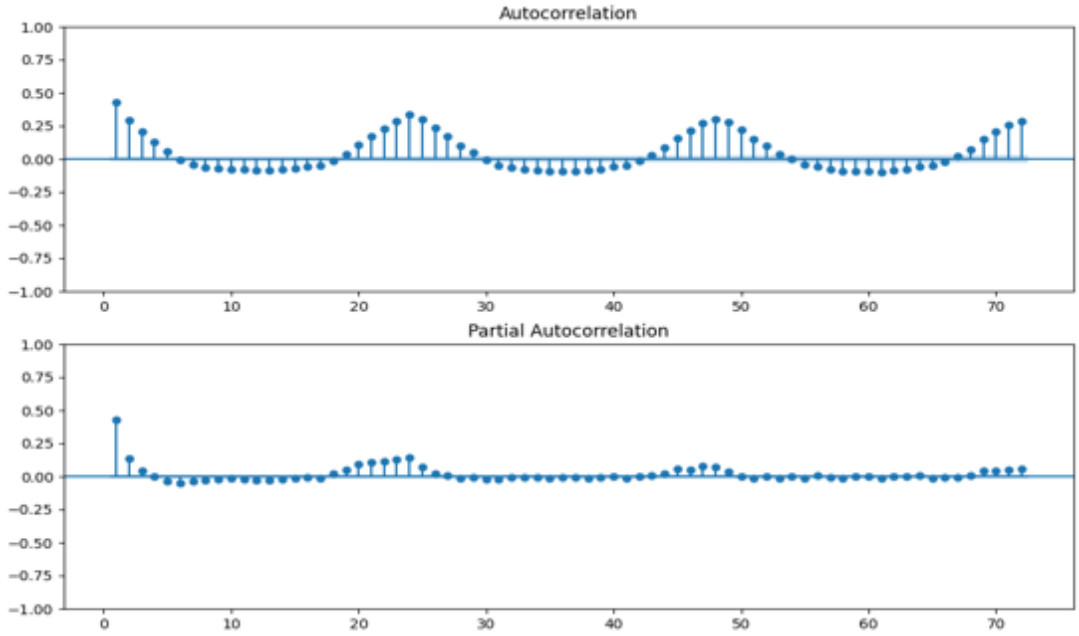

Figure 36: Auto and partial correlation for energy consumption.

As we already knew our series are seasonal and our ACF plot confirms this pattern. First significant lag is lag 1. The energy consumption raises/decreases, depends on the hour of day, gradually from hour to hour. Hence the energy value during the previous hour might tell us something about energy during the current hour. Next important lag is 24. 24-hour lag shows that energy consumption today at 4pm might hint about energy consumption tomorrow at 4pm. With PACF we can see that lags 1 and 24 have the highest correlation. This means that seasons 24 hours apart are directly correlated regardless of what is happening in between.

4.5. Evaluation Metrics

For assessing the performance of machine learning models used for load forecasting, we are going to use the following evaluation metrics based on error calculation. The most common used evaluation metrics in the literature [101] around this area is Root Mean Square Error (RMSE). RMSE calculates the standard deviation of the prediction errors to indicate the way the predicted data is clustered across the best fit line. Low values of RMSE shows that the model is more accurate in forecasting the load [101]. The mathematical formula is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{4.5-1}$$

where n is the number of outputs, $\hat{y} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n\}$ is the output of the forecasting model and $y = \{y_1, y_2, \cdots, y_n\}$ is the actual value corresponding to the forecasting result.

4.6. Implementation

All models have been developed using Keras API on top of TensorFlow [103]. The machine used for the analysis has the following specifications:

- ❖ Processor: 2.6 GHz 6-Core Intel Core i7
- ❖ Memory: 32 GB 2400 MHz DDR4
- ❖ Graphics: Radeon Pro 560X 4 GB, Intel UHD Graphics 630 1536 MB

The experiments were run on CPU, due to complex setup of enabling GPU processing for this specific graphics card.

There models had been applied on individual house data; SARIMAX, Vanilla-LSTM and Encoder-Decoder LSTM. For all models, last month (August 2020) has been kept for validation and the rest (August 2018 – July 2020) were used for training.

4.6.1. SARIMAX

Parameter estimation is the process of specifying the parameters (see 3.1.2) of a SARIMAX model. A good understanding of the data and the context they represent is crucial to identify the right parameters. There are modern/automated approaches like **grid-search** and **auto.arima()** for that, however in our case we used Box–Jenkins (see 3.1.3) method instead. Box–Jenkins forces you to deep dive in the problem and not brute force the way to the parameters. Let's estimate the parameters based on our theoretical understanding of ACF and PACF plots (see 4.4.5)

Estimates:

- **s**: In ACF plot there is one peak and one valley every 24 hours. Thus, seasonal period could the set to **24**.
- **p**: In ACF plot y_{t-1} is the first significant lag. We also notice that there is a gradual change where y_{t-1} is not drastically different from y_t, hence trend autoregressive order will be set to **1**.
- **d**: Using stationary check performed in 4.4.4 we could set trend differencing to **1.**
- **q**: Based on PACF correlations we can set moving average order to **1,** since its the most significant lag.
- **P**: Setting seasonal autoregressive order to **2** will allow us to use the first and second seasonally offsets (24) in the model.
- **D**: Using stationary check performed in 4.4.4 we can use first degree seasonal differencing **1.**
- **Q**: As shown in our PACF graph first lag have a significant correlation; hence seasonal moving average will be set to **1**.

We concluded on the following model:

$$SARIMA(1,1,1) \times (2,1,1)_{24}$$

4.6.2. Vanilla LSTM

A Vanilla LSTM is an LSTM model with a single hidden layer of LSTM units and an output layer, which is used to make the prediction. Our model is shown in the following code snippet:

```
units = 256
initializer = tf.keras.initializers.TruncatedNormal(mean=0.0,
stddev=0.5, seed=22)
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.LSTM(units,
input_shape=(x_train.shape[1], x_train.shape[2]),
kernel_initializer=initializer))
model.add(tf.keras.layers.Dense(units=args.horizon))
model.compile(loss='mean_squared_error', optimizer='adam')
```

In this case, we define a model with 256 LSTM units in the hidden layer and an output layer that predicts a horizon, 24 hours in this work. An output/dense layer is a fully connected layer that helps in changing the dimensionality of the output from the preceding layer. The model is fit using the efficient "adam" [104] version of stochastic gradient descent and optimized using the mean squared error, or "mse" loss function. The model expects the input shape to be three-dimensional with [samples, timesteps, features], therefore, train data have been reshaped before fitting them to the model. After trial and error, we ended up using also a kernel initializer. Initializers define the

way to set the initial random weights of Keras layers. TruncatedNormal initializer generates tensors with a normal distribution but values that are more than two standard deviations from the mean are discarded and re-drawn.

Keras utils includes a method to have a visual representation of the defined model. Figure 37 below depicts our Vanilla LSTM. The initial input is the past 24 hours of consumption along with the weather and hour of day features (8 in total). The final output is the prediction of consumption for the next 24 hours.

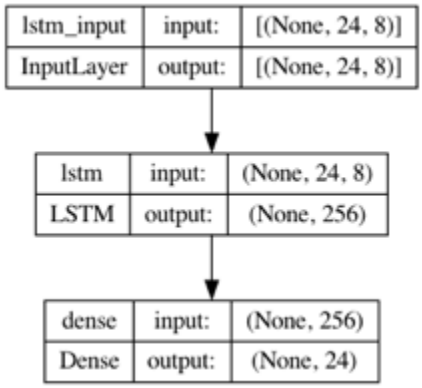

Figure 37: Vanilla LSTM model plot.

The code snippet below represents the fitting command. Keras supports early stopping of training via a callback called EarlyStopping. This callback allows you to specify the performance measure to monitor, the trigger, and once triggered, it will stop the training process. In our case we monitor the loss metric as decreases. A delay to the trigger in terms of the number of epochs on which we would like to see no improvement is configured. This can be done by setting the patience argument. However, we do not want to exit too early, so we allow the network to run at least for half the number of epochs.

```
# This will be used to avoid overfitting
es = tf.keras.callbacks.EarlyStopping(monitor='loss', mode='min',
verbose=1, patience=10, start_from_epoch = int(500/2.0))
history = model.fit(x_train, y_train, epochs=500, batch_size=32, verbose=2,
shuffle=False, callbacks=[es])
```

4.6.3. Encoder-Decoder LSTM

Encode-decoder architecture is quite popular for sequence-to-sequence forecasting problems. It consists of at least two RNN/LSTMs, where one acts like an encoder while the other as decoder. The main task of encoder is to read and interpret the input. On top, it compresses the input to a fixed-length vector and passes that to the next level. A RepeatVector layer is used to repeat the context vector for a number of future steps (24 in our case) and then it is fed to the decoder part. The decoder performs the forecasting and passes the output, to a fully connected Dense layer is applied to each time step via TimeDistributed wrapper, so separates the output for each time step.

```
n_timesteps, n_features, n_outputs = x_train.shape[1], x_train.shape[2],
y_train.shape[1]
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.LSTM(200, input_shape=(n_timesteps, n_features)))
model.add(tf.keras.layers.RepeatVector(n_outputs))
model.add(tf.keras.layers.LSTM(200, return_sequences=True))
model.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(100)))
model.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(1)))
model.compile(loss='mse', optimizer='adam')
```

As we can see in the code snippet above, we define an encoder LSTM layer with 200 units, which reads the input and will output a 200 element vector. The input consists of sequences of 24 hour consumption along with weather data and the hour of day. Input sequence is repeated multiple times, once for each time step; hence the RepeatVector layer. The decoder layer is also defined with 200 units and it is important to output the entire sequence; return_sequences=True. This means that each of the 200 units will output a value for each of the 24 hours. After the decoder, we added two dense layers wrapped in TimeDistributed layer. In that way, the model is

able to extract the context of each time step and interpret each time step separately by reusing the same weights. Figure 38 helps comprehending the process providing a visual representation.

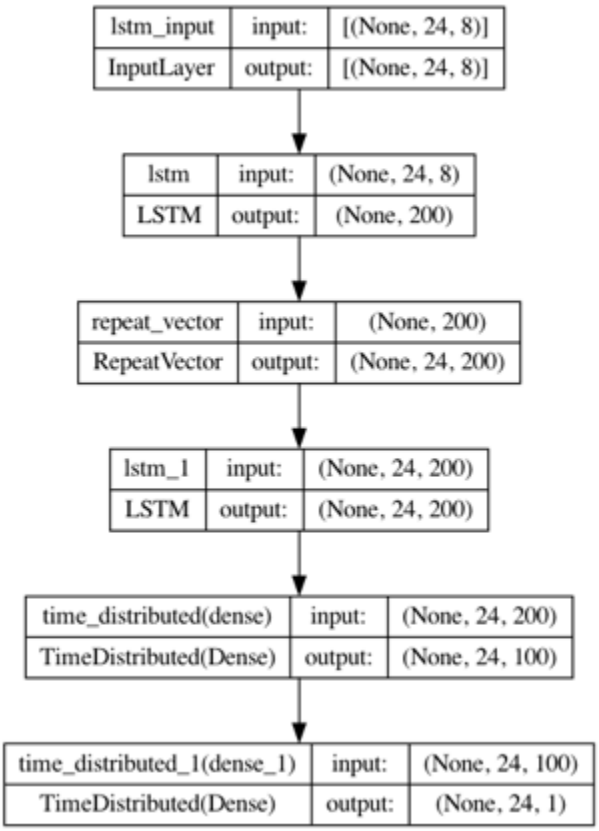

Figure 38: Encoder-Decoder LSTM model plot.

Hyperparameter of fitting the models remained the same as the Vanilla LSTM.

```
# This will be used to avoid overfitting
es = tf.keras.callbacks.EarlyStopping(monitor='loss', mode='min',
verbose=1, patience=10, start_from_epoch = int(500/2.0))
history = model.fit(x_train, y_train, epochs=500, batch_size=32, verbose=2,
shuffle=False, callbacks=[es])
```

5. Results

In this section we are going to present the results of our experiments and pinpoint useful learnings. The introductory idea was to create a unified model based on aggregated data; 80% (46 houses) of the houses were considered in the aggregated dataset by calculating the media of those houses per hour. 5 houses were kept unseen in order to validate the model. A Vanilla LSTM model was fitted in the aggregated dataset and then we applied it on the unsees houses 2 and 3. Results are shown in Figure 39 below.

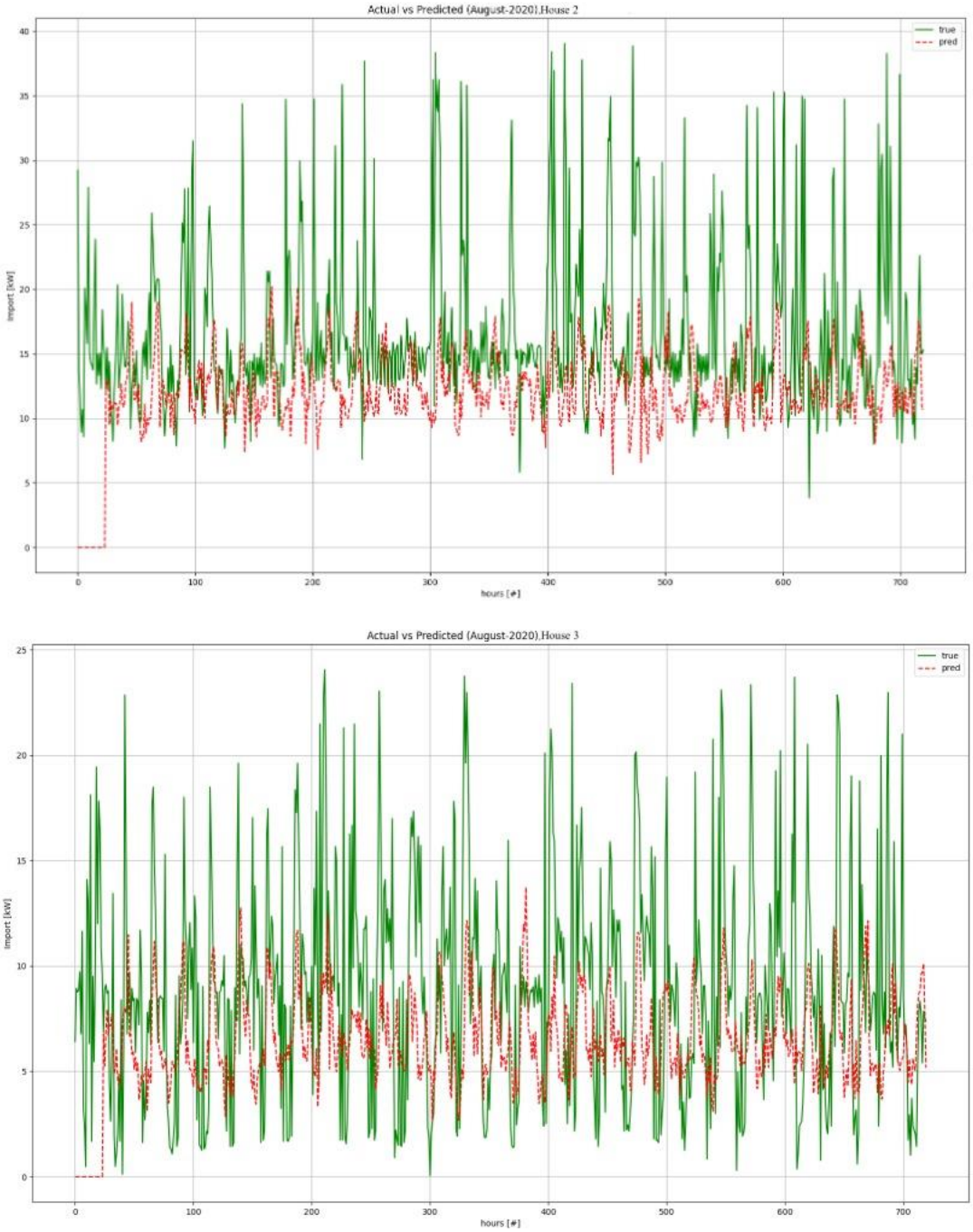

Figure 39: Vanilla LSTM on aggregated model applied on houses 2 and 3.

It is obvious that the model is not good since it is not able to follow the hourly pattern of neither of the two houses. The model does not follow neither the peaks nor the valleys, it predicts values close to a specific mean range. Specifically, for house 2 it constantly underestimates the consumption and it does not predict values higher than 20kW. On house 3 the model is a bit better and at least it tries to follow the hourly pattern. The main reason for this behavior is that the mean consumption of house 3 is closer to the aggregated dataset mean consumption ~8kW. Aggregating data is like collapsing multiple houses to one; hence if we want to get to that direction the houses that are part of the aggregation should follow a similar load pattern.

Since the model cannot be generalized, we decided to create a model for each house. Our dataset consists of 57 houses so the idea is to create respective models. For production environments this approach might be not ideal since we need to maintain/improve/deploy many models. In this work, we created models for houses 2, 3 and 69.

5.1. SARIMAX

Figure 40 below depicts the diagnostics of SARIMAX model for house 2 (top) and house 3 (bottom). The standardize residuals (top left) don't display any obvious patterns over time. They appear as white noise. The Normal Q-Q (bottom left) indicates that the ordered distribution of residuals follows the linear trend of the samples taken from a standard normal distribution with N(0, 1). However, the slight curving indicates that our distribution is heavy tailed. This pattern seems a bit smoother in house 69.

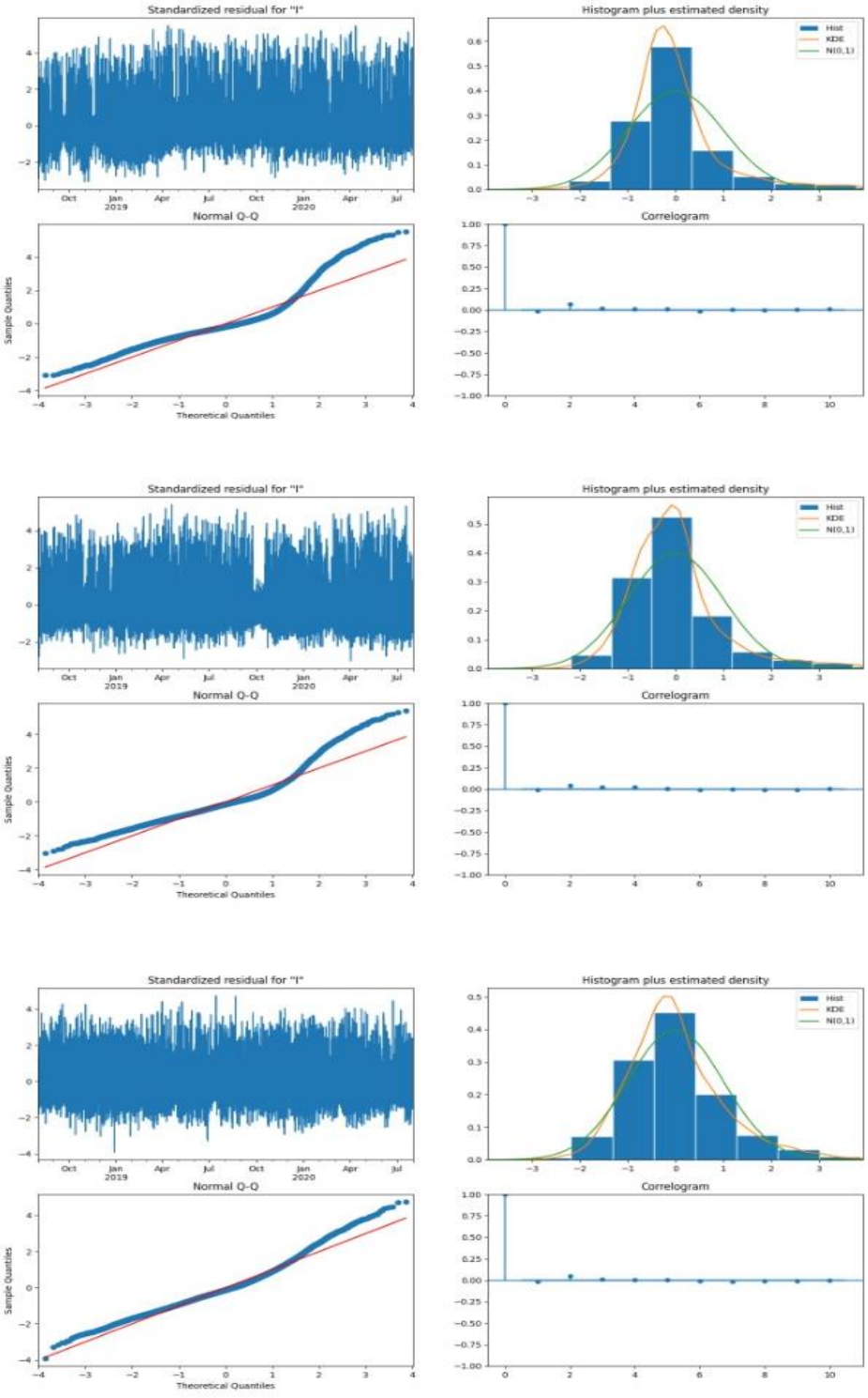

Figure 40: SARIMAX model diagnostics for house 2 (top), house 3 (middle) and house 69 (bottom) .

A heavy tailed distribution has a tail that's heavier than an exponential distribution. In other words, a distribution that is heavy tailed goes to zero slower than one with exponential tails. Heavy tailed distributions tend to have many outliers with very high values. The heavier the tail, the larger the probability that you'll get one or more disproportionate values in a sample.

Histogram and estimated density (top right) shows that Kernel Density Estimation (KDE) follows the N(0,1) line however with noticeable differences. As mentioned before our distribution has heavier tails. The Correlogram (bottom right) shows that the time series residuals have low correlation with lagged versions of itself.

Figure 41, depicts the result of applying house2 SARIMAX model on test data. RMSE is ~7.9kW. The top plot indicates that the model does not follow the real pattern that well. The model shows a strong seasonality and cannot follow steep peaks. RMSE is a qualitative metric and in cases like this considering only that might be misleading. Thus, it is important to have a visual representation where we actually see how the model behaves.

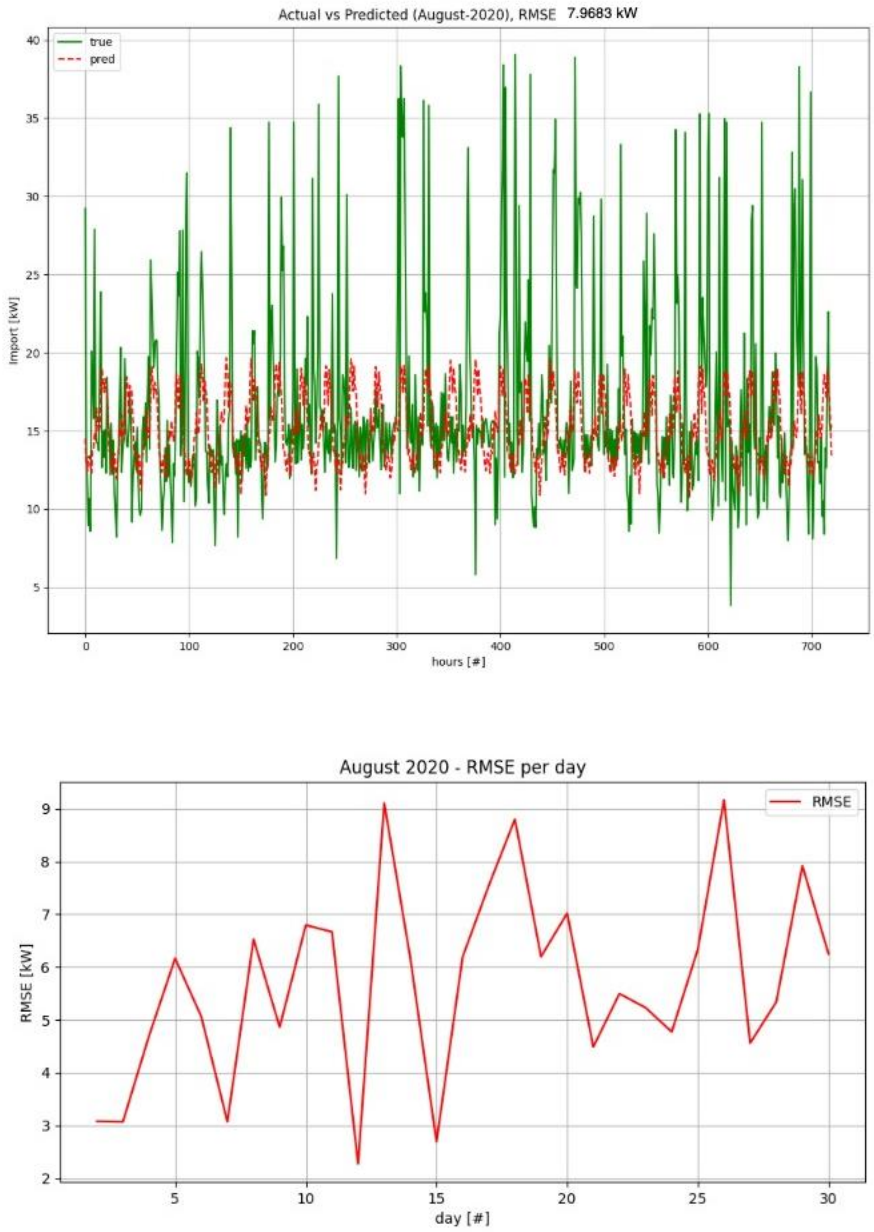

Figure 41: SARIMAX on house 2 (top). RMSE per day (bottom).

Figure 42 depicts the result of applying house3 SARIMAX model on test data. RMSE is low ~5.6kW. The model tends to follow the valleys but does not keep up with peaks, proving that SARIMAX is quite sensitive in outliers. Hardest days to predict were day 9 and day 14, with an RMSE ~6.6kW and ~6.8 respectively.

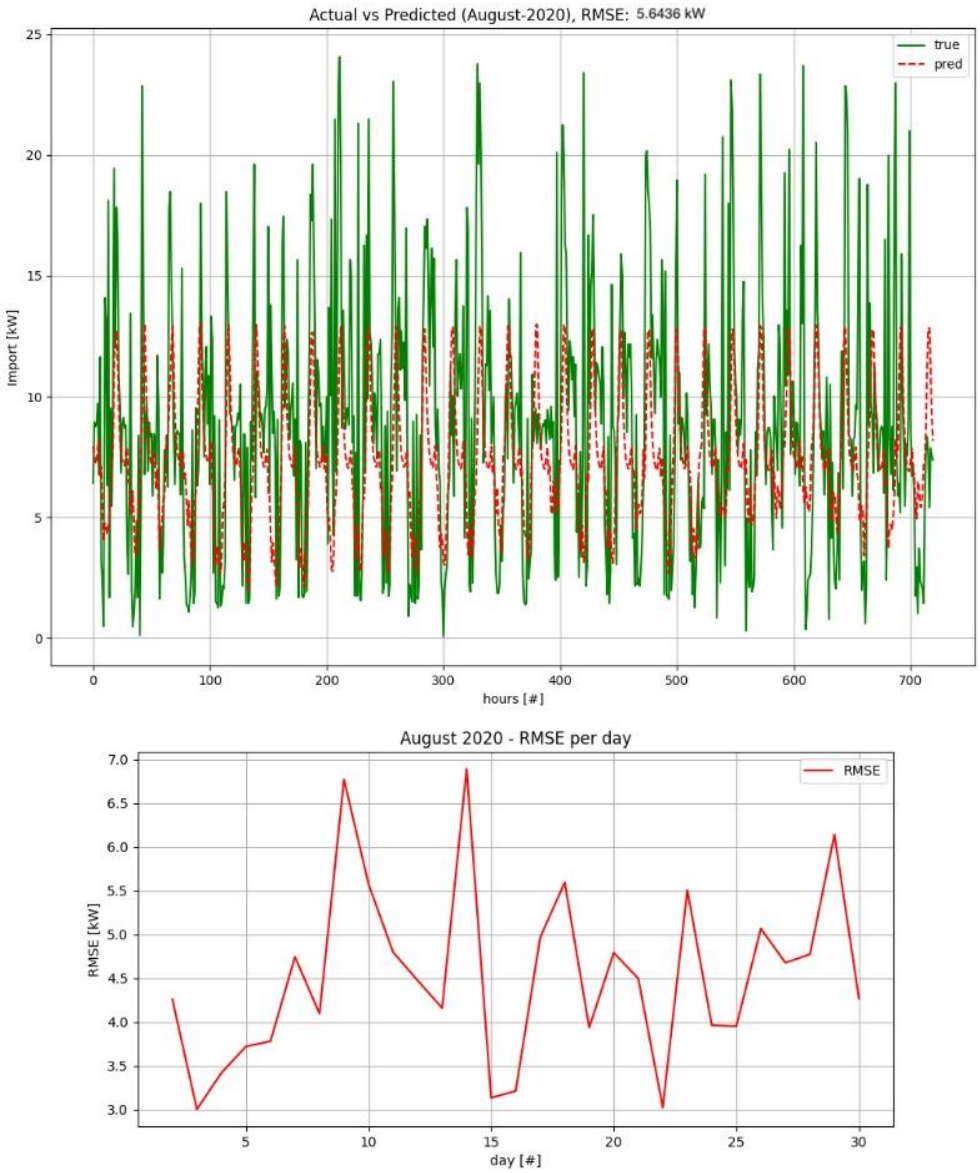

Figure 42: SARIMAX on house 3 (top). RMSE per day (bottom).

Figure 43 below visualizes how SARIMAX model of house 69 performs. House 69 proves to be more challenging, thus the RMSE is ~15kW. As in previous cases, the model is able to predict low values, but it is unable to predict high consumption and constantly underestimates. Highest RMSE value occurs in day 17.

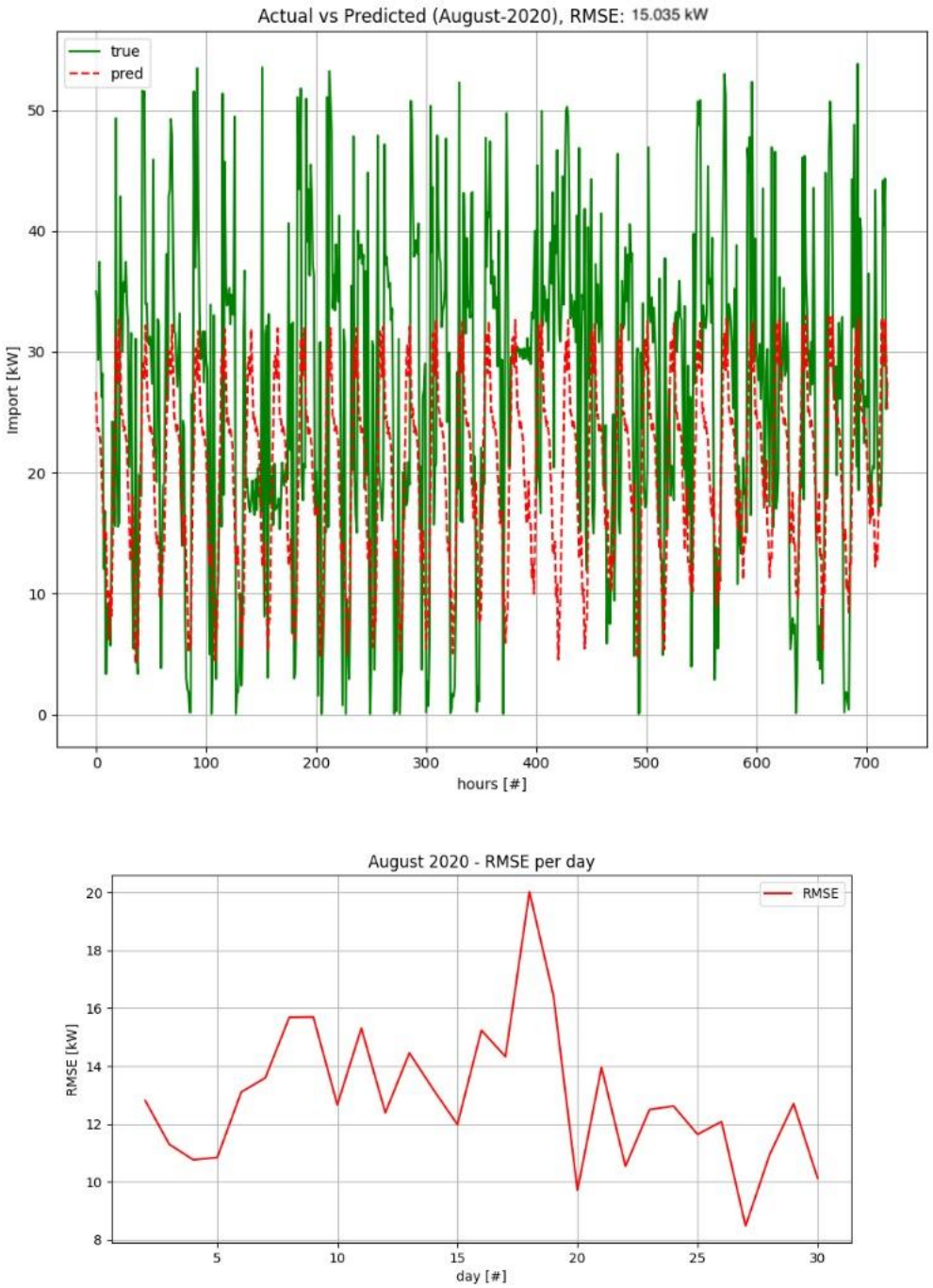

Figure 43: SARIMAX on house 69 (top). RMSE per day (bottom).

5.2. Vanilla LSTM

Figure 44 top part depicts the real consumption (green line) vs the predicted one (red dotted line) for August 2020. The mean RMSE of the month is 7.4kW. August 1st (first 24 hours in the top figure) is not predicted since our model needs the previous 24 hours as an input; hence the prediction for that day is 0. This specific house has no deep valleys but instead has some high peaks. The reason the smart meter registers those values is unknow and needs more characteristics (business context, property's square meters, residents' demographics) to conclude. We notice though that the model tries to follow those peaks.

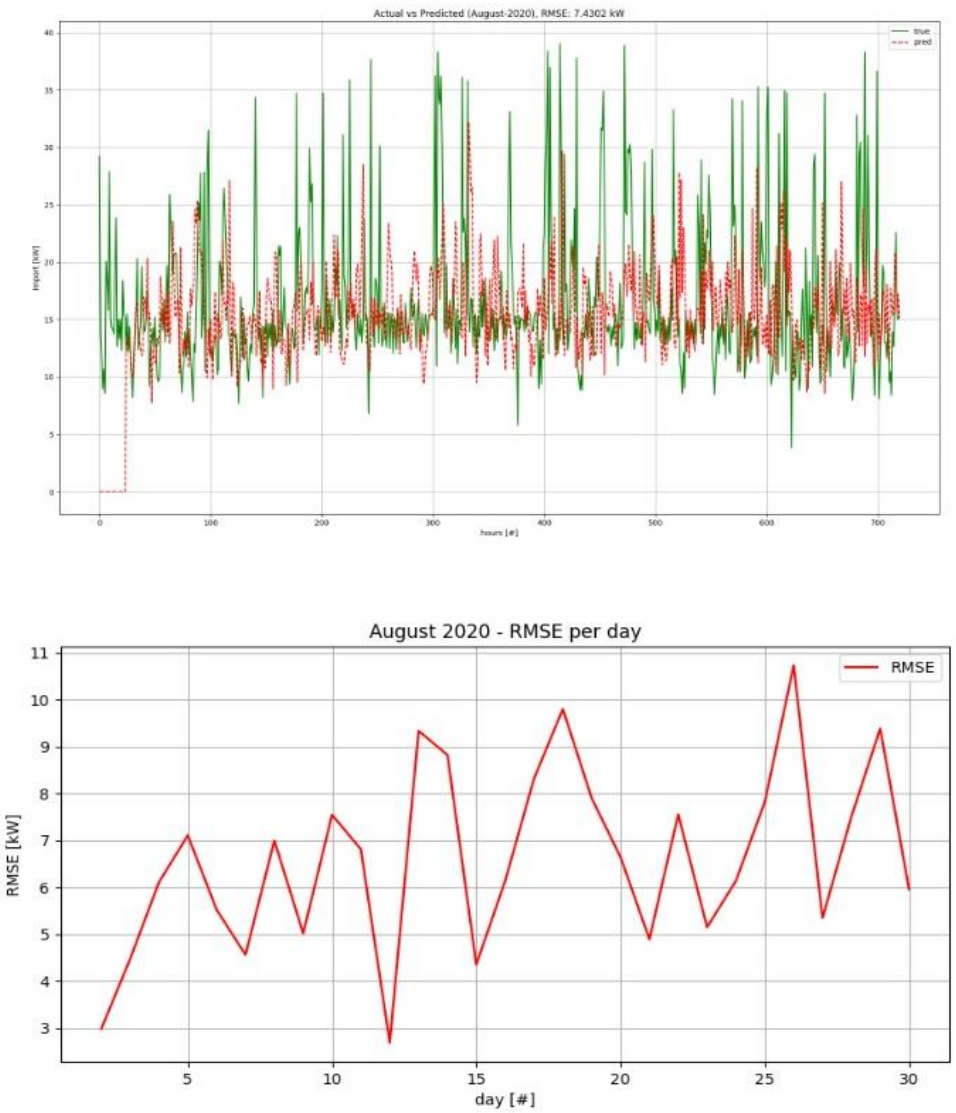

Figure 44: Vanilla LSTM on house 2 (top). RMSE per day (bottom).

The bottom part of the figure shows the RMSE per day. We notice that days 2, 7, 9, 12, 15, 21 have quite low RMSE <=5kW. On the contrary, there are days that the error is high, 13, 17, 29, and reaching the maximum ~11kW on day 26.

As shown in Figure 45, model of house 3 behaves quite well; this argument has been supported also from RMSE's value, 5.2kW. It seems that it does not have much trouble following both peaks and valleys. Some of the peaks though are quite steep and the model cannot predict them well. This pattern should be investigated further, because it is not very common to have such a high difference between two or three consecutive hours.

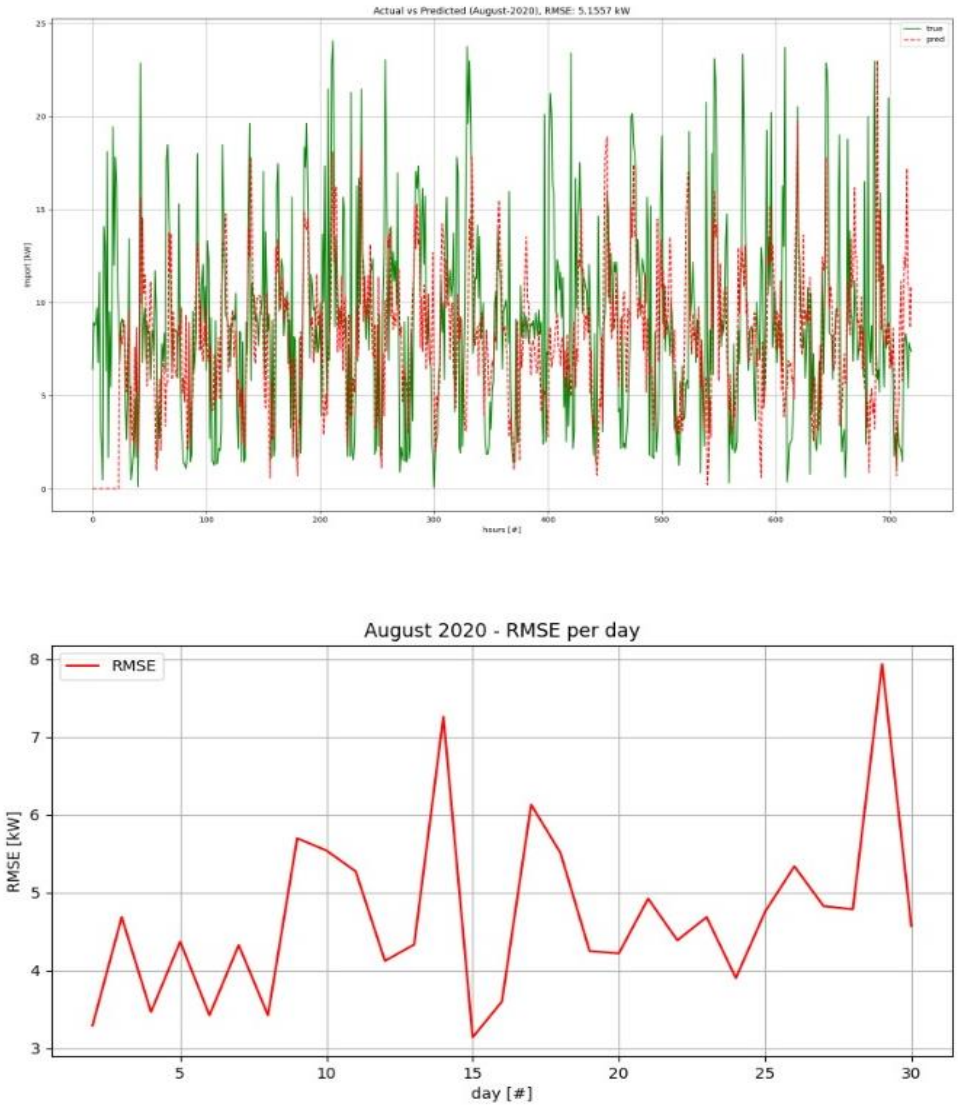

Figure 45: Vanilla LSTM on house 3 (top). RMSE per day (bottom).

Looking both plots in combination we notice that some hours are harder to predict due to these high peaks. For instance, in day 9, hours 193 – 216, the error is higher than 7kW. Similarly, on day 17 and 29 the model has hard time and the error is 6kW and 8kW respectively.

House 69, Figure 46, proved to be more challenging. The RMSE is significantly higher than the other two houses, 14.5kW. There is a mixture of good days/range of hours and bad ones where the model underestimates load consumption. This is an indication that model would probably need more units, however due to the incapability of running on GPU that wasn't possible in our case.

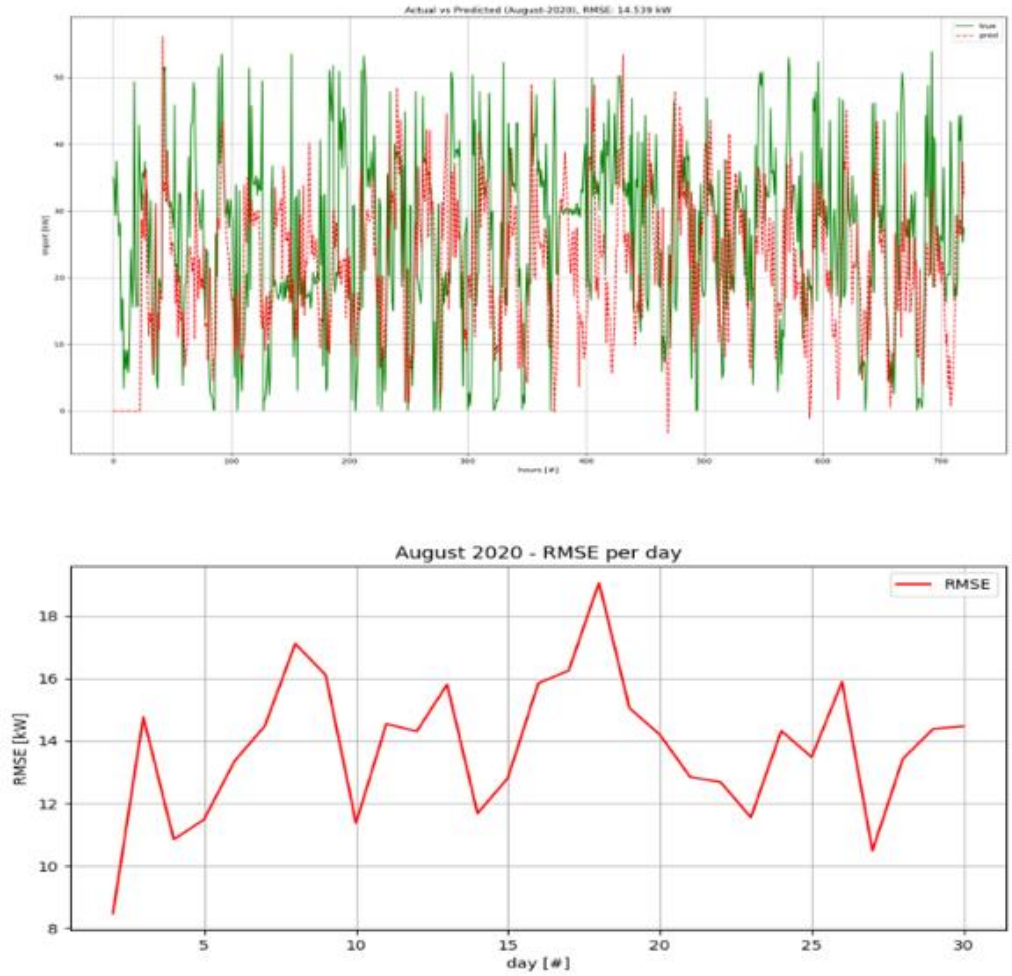

Figure 46: Vanilla LSTM on house 69 (top). RMSE per day (bottom).

A series of days have RMSE >=15kW: 8, 9, 13, 16, 17, 18, 19, 26; proving in combination with top plot that model underestimates.

5.3. Encoder-Decoder LSTM

Figure 47 depicts encoder-decoder LSTM model for house 2. First thing to notice is that RMSE is slightly lower from 7.4kW to 7.3kW. The model is relative realistic in its predictions; however, it has a bit the tension to overestimate. For instance, take a closer look on hours range 650-680 and 710-720, the model there predicts higher for a sequence of hours than the actual consumption values.

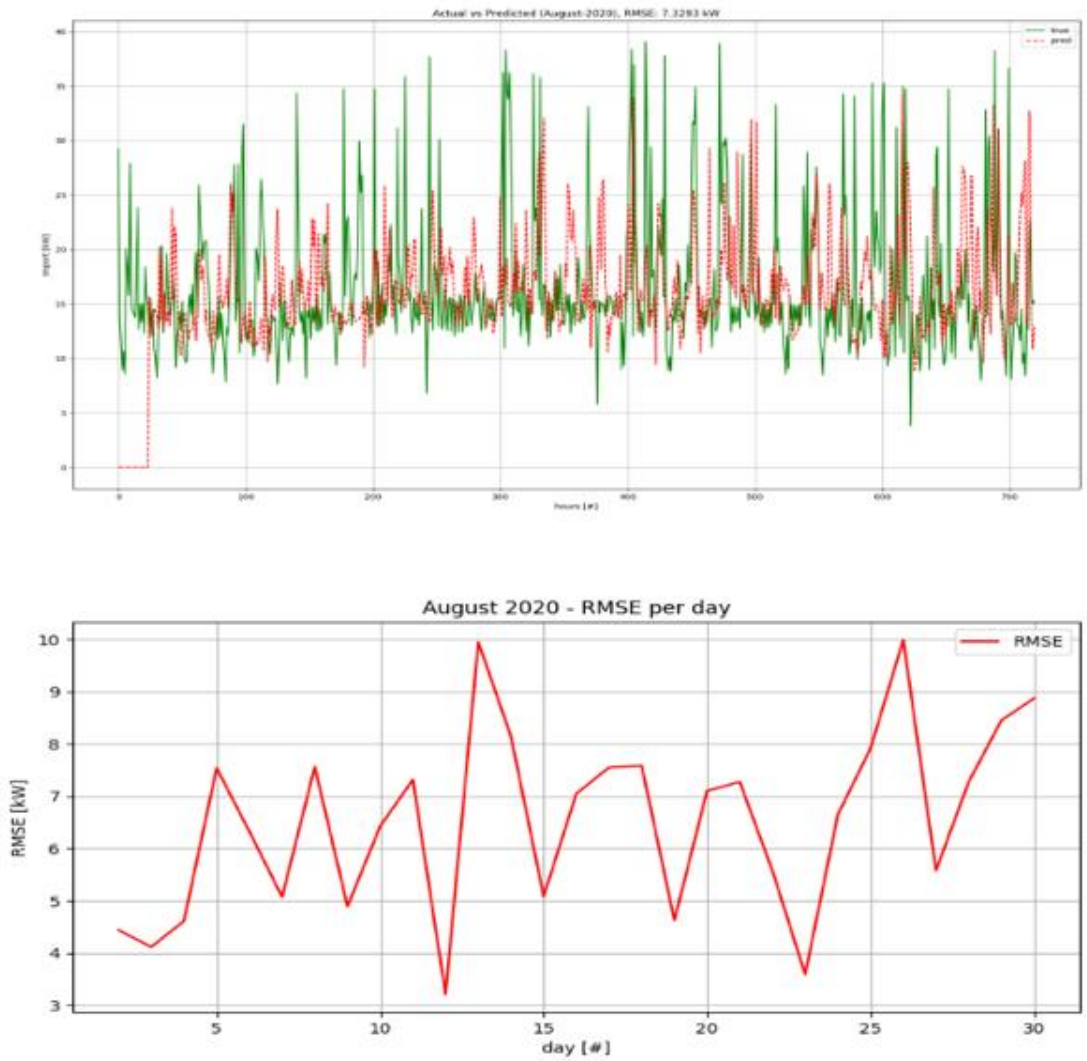

Figure 47: Encoder-Decoder LSTM on house 2 (top). RMSE per day (bottom).

Comparing the RMSE per day with the respective one from Vanilla LSTM model we notice that it is more balanced and has only to steep peaks; days 25 and 26, hence the error is more spread out throughout the month.

Next, Figure 48, is related to house 3 where the RMSE is a bit higher in this case 5.3kW. The model behaves more or less the same way as Vanilla LSTM and follows the consumption pattern efficiently. It underestimates a bit the high peaks, a pattern that occurs in the previous model as well.

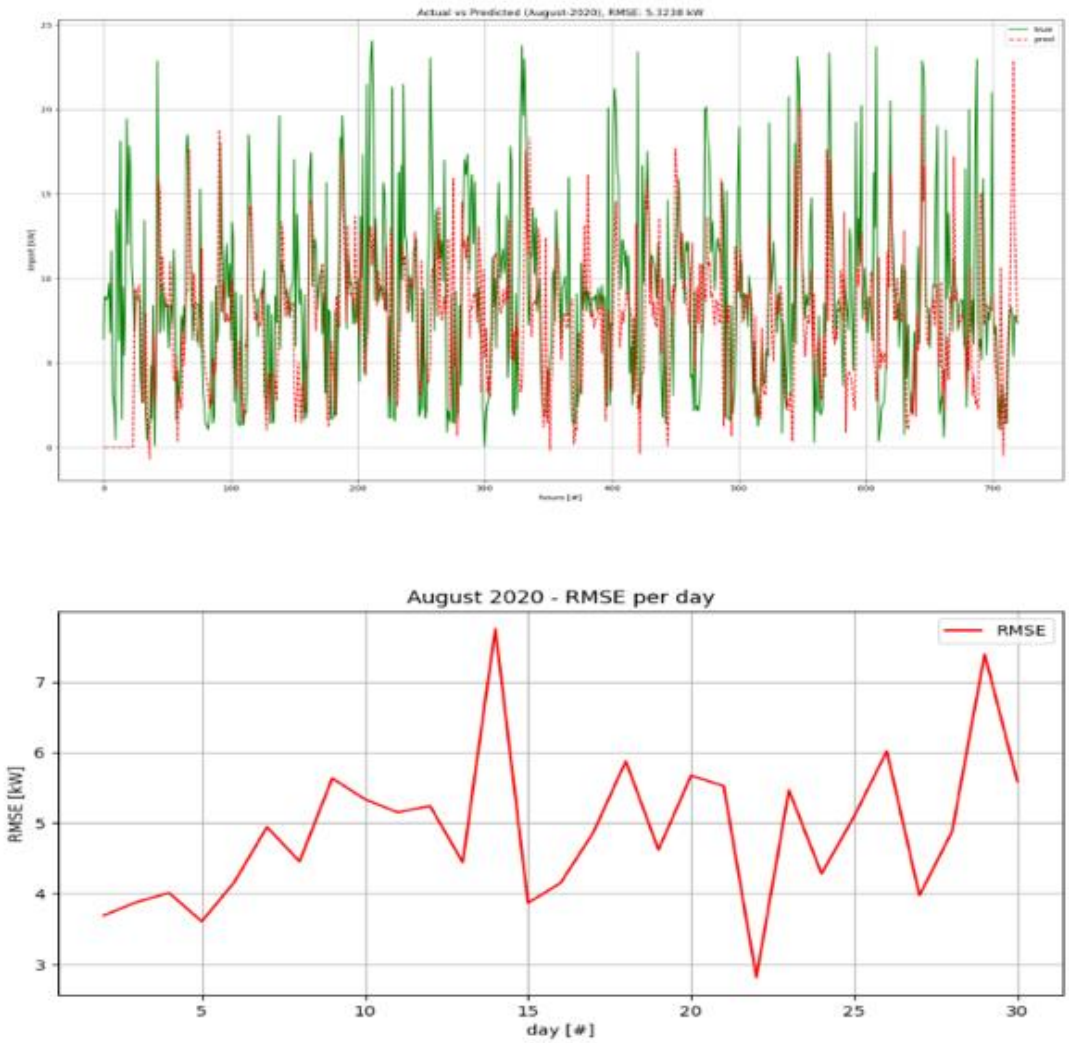

Figure 48: Encoder-Decoder LSTM on house 3 (top). RMSE per day (bottom).

Judging from RMSE per day figure the model has hard time predicting values in days 14 and 29. A pattern we noticed also in the previous model.

Figure 49, visualizes the pattern and the respective errors for house69. The performance didn't change much comparing with the previous model. The mean RMSE remained more or less the same ~14.5kW, same the prediction pattern. The model mostly underestimates consumption. The highest RMSE occurs in day 9 with a value of 20kW.

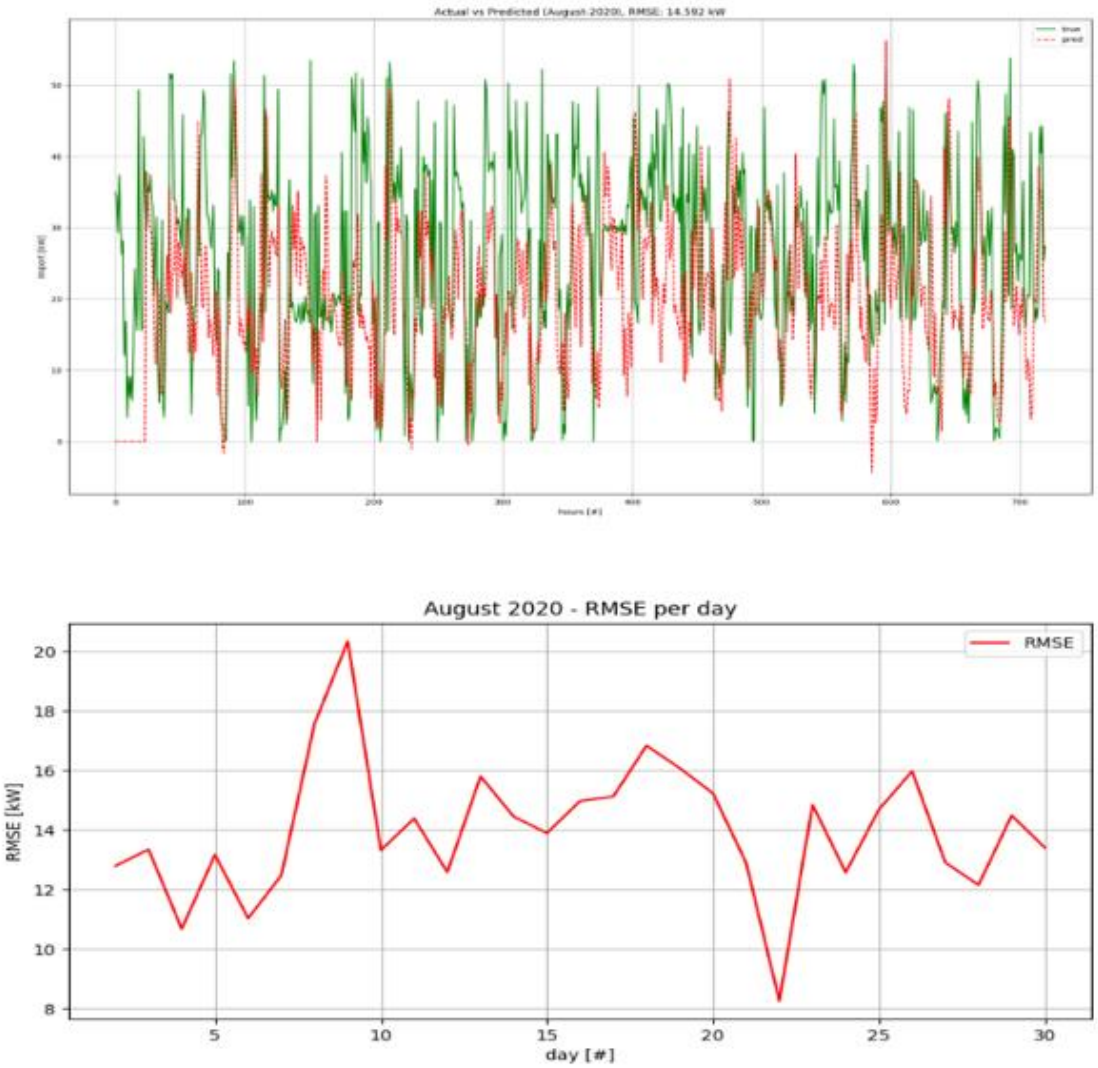

Figure 49: Encoder-Decoder LSTM on house 69 (top). RMSE per day (bottom).

5.4. Apply model on unseen houses

We performed some more experiments applying the Vanilla LSTM house models on unseen houses. The main idea is to apply an existing model on a house that has a similar consumption pattern, Table 4. House 2 model is applied on house 66.

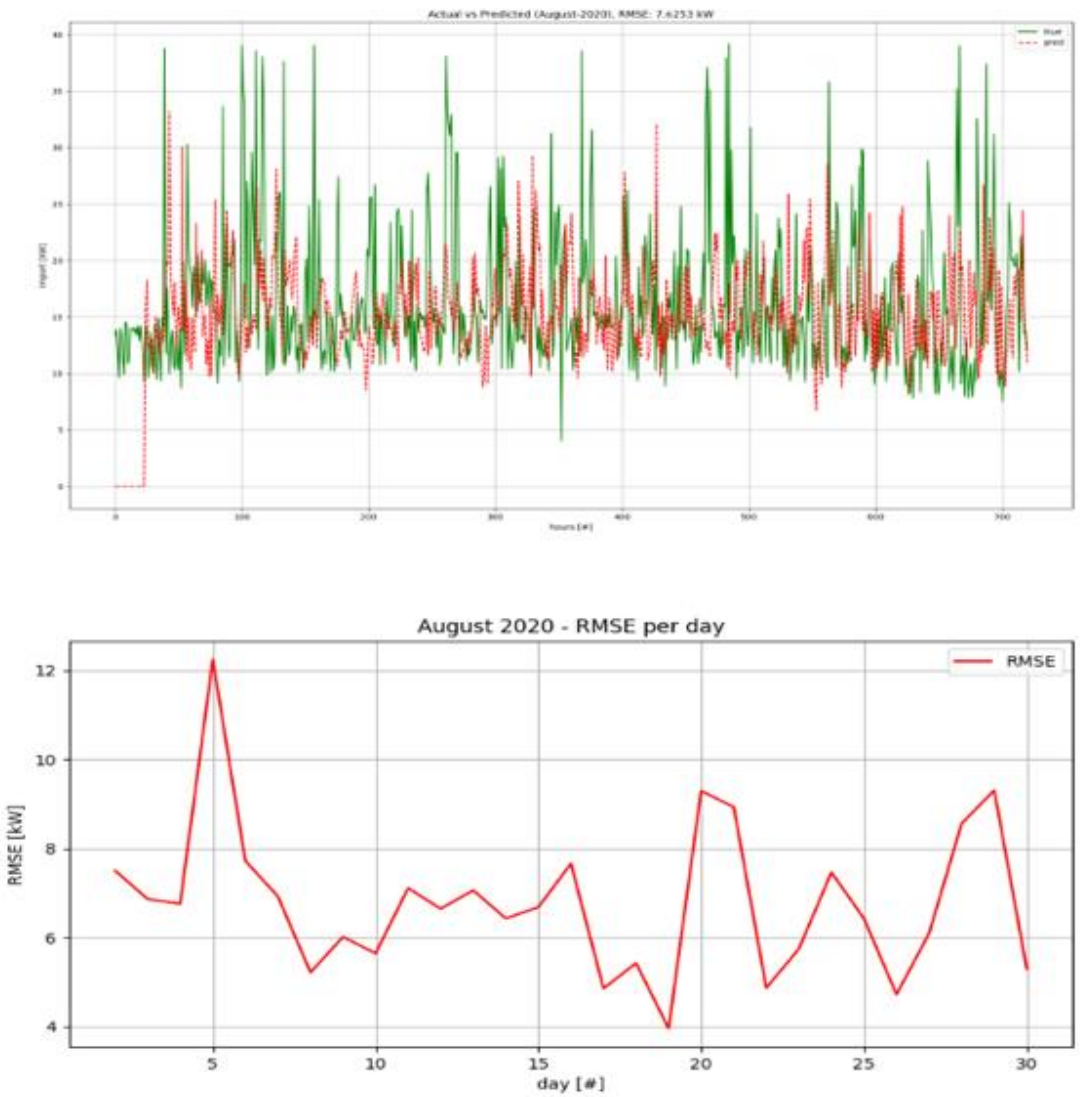

Figure 50:Vanilla LSTM house 2 model applied on house 66 (top). RMSE per day (bottom).

Figure 50 above, shows the result of applying house 2 vanilla LSTM model on house 66. From the top graph we could notice that consumption pattern of house 66 is similar to house 2; hence the model performs well on this "unseen" house. RMSE is only a bit higher ~7.6kW from ~7.4kW of house 2. Only exception is day 5 where the RMSE peaks on ~12kW.

Another experiment is shown in Figure 51 below, we applied vanilla LSTM house 3 model on house 4. As we can see in most cases the model follows the pattern even though it hasn't been trained on house 4 data. However, it is again obvious that the consumption pattern is similar. RMSE is a bit higher ~5.2kW. There are some hour ranges that the model mostly overestimates, e.g., 160-180, 210-230, 670-685, but in general the RMSE per day stays low.

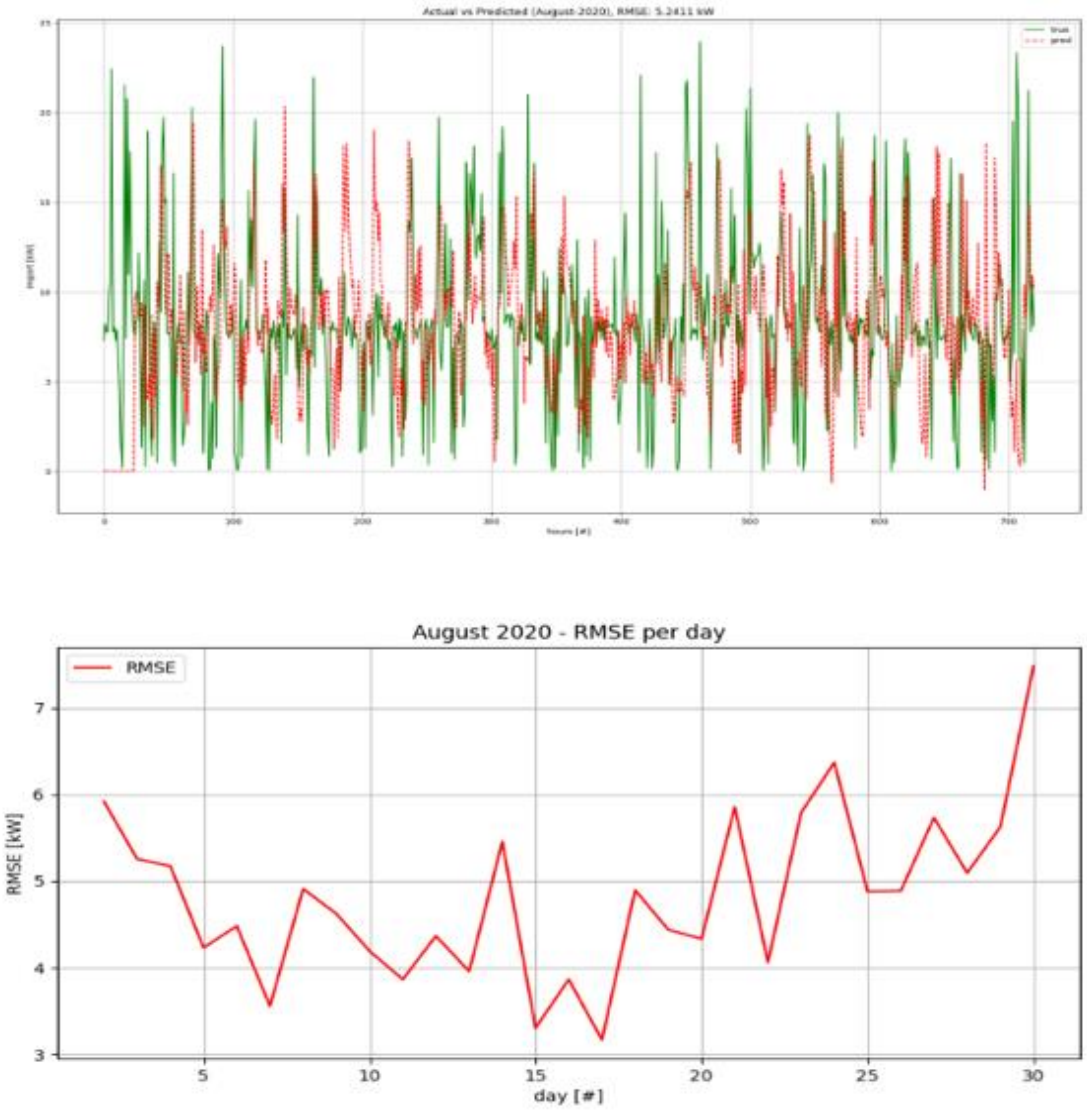

Figure 51: Vanilla LSTM house 3 model applied on house 4 (top). RMSE per day (bottom).

6. Conclusion & Future Work

This work started with an overview on smart grids. Transforming the existing grid to a smart one brings many challenges, but it is important for scaling and incorporating renewable energy sources. ICT technologies provide all the necessary tooling for a step-by-step transition to a smart grid.

Next topic was to study short-term load forecasting for residential users. It is a quite challenging problem considering the versatility of consumption in a house. Smart meter data were obtained from [99], consisting of consumption data of 77 individual houses for the period August 2018 – August 2020. After data retrieval, we performed data pre-processing, data cleaning and explanatory data analysis, in order identify patterns and draw conclusions. Initially, we attempted to create a unique model that could fit all houses. However, this approach proved to be wrong, see 5. Aggregated data set is like collapsing the houses to 1, judging from the result we see that prediction cannot follow an unseen house's pattern. It cannot follow neither the peaks nor the valleys. It predicts values close to a specific mean range. In our use-case then it is best to create a model for each house or one model per cluster of houses (similar load pattern).
Thereafter, we created separate models, SARIMAX, Vanilla-LSTM, Encoder-Decoder LSTM, and presented the results. An overview can be seen in the following table.

Table 5: Summing up results of short-term load forecasting.

House id	RMSE SARIMAX	RMSE Vanilla LSTM	RMSE Enc-Dec LSTM
2	7.9683	7.4302	7.3293
3	5.6436	5.1557	5.3238
4 (unseen)	-	5.2411	-
66 (unseen)	-	7.6253	-
69	15.035	14.539	14.701

It is very important to combine the calculation of RMSE with a plot of true value vs predicted. We noticed also that models in some cases cannot follow the pattern and report wrong values, the following reasons might give an explanation:

- There might be the case that we have erratic values (ground truth) due to malfunctioning smart meters; hence the predicted value could be actually close to reality.
- Acceptable model error within a reasonable boundary. Acceptable should always be defined by the business.

Another thing to consider, is that SARIMAX models are quite large ~14GB compared to LSTM ~6MB. Hence, they cannot be deployed in a resource constraint environment, like raspberry pi.

For future work , we would like to perform the following actions:

- Apply more sophisticated ways for replacing missing values, e.g., consider the hour of the previous day, replace long periods using distributions. In a production environment, it is unlikely to have long periods of missing data since alerts have been set and engineers fix the issues.
- Incorporate more features like EXPORT_KW, PV_KW, BATTERY_KW.
- Apply a classification algorithm, e.g., K-Nearest Neighbor (k-NN) [105], for grouping houses with similar patterns and then create models based on those profiles.
- Test Prophet and CNN/LSTM Encoder – Decoder.

The following saying sums up what we have experienced throughout this work.

"The most reliable way to forecast the future is to try to understand the present."

— John Naisbitt

References

[1] Hobbs, B. F., Jitprapaikulsarn, S., Konda, S., Chankong, V., Loparo, K. A., & Maratukulam, D. J. (1999). Analysis of the value for unit commitment of improved load forecasts. IEEE Transactions on Power Systems, 14(4), 1342–1348.

[2] Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2019). Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Transactions on Smart Grid, vol. 10, pp. 841-851.

[3] Espinoza, M., Suykens, J. A. K., Belmans R., & De Moor B. (2007). Electric Load Forecasting. IEEE Control Systems, 27(5), 43–57.

[4] Salman, S. K. (2019). Evolution of Conventional Power Systems to Smart Grids. 2019 54th International Universities Power Engineering Conference (UPEC), 1-6.

[5] Khoussi, S., & Mattas, A. (2017). A Brief Introduction to Smart Grid Safety and Security. Handbook of System Safety and Security, 225–252.

[6] Sioshansi, F. P. (Ed.). (2011). Smart grid: integrating renewable, distributed and efficient energy. Academic Press, 74-77.

[7] Shabanzadeh, M. & Moghaddam, M. (2013). What is the Smart Grid? Definitions, Perspectives, and Ultimate Goals. Power System Conference.

[8] Department of Energy, United States (DOE), "The Smart Grid: An Introduction", Washington, DC., 2003.

[9] Electric Power Research Institute, IntelliGrid Smart Grid Roadmap Methodology and Lessons Learned, Dec 31, 2012, available online: https://www.epri.com/research/products/1026747.

[10] Energy Independence and Security Act of 2007, Title XIII, Dec 19, 2007, available online: https://www.govinfo.gov/content/pkg/PLAW-110publ140/pdf/PLAW-110publ140.pdf.

[11] European Union Commission Task Force for Smart Grids, available online: https://www.cencenelec.eu/areas-of-work/cen-cenelec-topics/smart-grids-and-meters/smart-grids/.

[12] ABB, Smart grid solution overview, 2016, available online: https://library.e.abb.com/public/8e65184b492b40b9a3e1c6e84eb1bfd7/smart-grid-solution-overview.pdf.

[13] National Institute of Standards and Technology, NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0., January 2010, available online: https://www.nist.gov/system/files/documents/public_affairs/releases/smartgrid_interoperability_final.pdf.

[14] Colak, I. (2016). Introduction to smart grid. 2016 International Smart Grid Workshop and Certificate Program (ISGWCP).

[15] National Energy Technology Laboratory (NETL), Understanding the Benefits of the Smart Grid, June 2010, available online: https://netl.doe.gov/sites/default/files/Smartgrid/06-18-2010_Understanding-Smart-Grid-Benefits.pdf.

[16] European Renewable Energies Federation (EREF), Energy Atlas – Facts and Figured about renewables in Europe, April 2018, available online: https://www.boell.de/sites/default/files/energyatlas2018_facts-and-figures-renewables-europe.pdf.pdf.

[17] Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2012). A Survey on Cyber Security for Smart Grid Communications. IEEE Communications Surveys & Tutorials, 14(4), 998–1010.

[18] Alotaibi, I., Abido, M. A., Khalid, M., & Savkin, A. V. (2020). A Comprehensive Review of Recent Advances in Smart Grids: A Sustainable Future with Renewable Energy Resources. Energies, 13(23), 6269.

[19] Huang, H., & Savkin, A. V. (2017). An energy efficient approach for data collection in wireless sensor networks using public transportation vehicles. AEU - International Journal of Electronics and Communications, 75, 108–118.

[20] Potdar, V., Chandan, A., Batool, S., & Patel, N. (2018). Big Energy Data Management for Smart Grids—Issues, Challenges and Recent Developments. Computer Communications and Networks, 177–205.

[21] Chu, X., & Ilyas, I. F. (2016). Qualitative data cleaning. Proceedings of the VLDB Endowment, 9(13), 1605–1608.

[22] Guerrero, J. I., García, A., Personal, E., Luque, J., & León, C. (2017). Heterogeneous data source integration for smart grid ecosystems based on metadata mining. Expert Systems with Applications, 79, 254–268.

[23] Sigeru, O., Sara, R., Gabriel V., Pedro (2017). Distributed Computing and Artificial Intelligence. 14th International Conference. Spain: Springer International Publishing AG,

[24] Omatu, S., Rodríguez, S., Villarrubia, G., Faria, P., Sitek, P., & Prieto, J. (Eds.). (2018). Distributed Computing and Artificial Intelligence, 14th International Conference. Advances in Intelligent Systems and Computing, 87-93.
[25] Virgilio, R. D. (2017). Smart RDF Data Storage in Graph Databases. 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), 872–881
[26] Bruno, S., Dellino, G., Scala, M. L., & Meloni, C. (2018). A Microforecasting Module for Energy Consumption in Smart Grids. 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 1-6.
[27] Stefan, M., Lopez, J. G., Andreasen, M. H., & Olsen, R. L. (2017). Visualization Techniques for Electrical Grid Smart Metering Data: A Survey. 2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService), 165-171.
[28] Yu, M., & Hong, S. H. (2015). A Real-Time Demand-Response Algorithm for Smart Grids: A Stackelberg Game Approach. IEEE Transactions on Smart Grid, 879-888.
[29] Kirschen, D., & Strbac, G. (2004). Fundamentals of Power System Economic. Chichester, UK: John Wiley & Sons, Ltd, 39-43.
[30] Min, C.-G., & Kim, M.-K. (2017). Impact of the Complementarity between Variable Generation Resources and Load on the Flexibility of the Korean Power System. Energies, 10(11), 1719.
[31] Yang, Y., Wang, M., Liu, Y., & Zhang, L. (2018). Peak-off-peak load shifting: Are public willing to accept the peak and off-peak time of use electricity price? Journal of Cleaner Production, 199, 1066–1071.
[32] Tang, Q., Yang, K., Zhou, D., Luo, Y., & Yu, F. (2016). A Real-Time Dynamic Pricing Algorithm for Smart Grid With Unstable Energy Providers and Malicious Users. IEEE Internet of Things Journal, 3(4), 554–562.
[33] Wang, Y., & Li, L. (2016). Critical peak electricity pricing for sustainable manufacturing: Modeling and case studies. Applied Energy, 175, 40–53.
[34] Abo-Khalil, A. G., Abdelkareem, M. A., Sayed E. T., Maghrabie H. M., Radwan A., Rezk H., & Olabi A. G. (2022). Electric vehicle impact on energy industry, policy, technical barriers, and power systems. International Journal of Thermofluids, Volume 13.
[35] Khan, B., Getachew, H., & Alhelou, H. H. (2021). Components of the smart-grid system. Solving Urban Infrastructure Problems Using Smart City Technologies, 385–397.
[36] Martins, J. F., Pronto, A. G., Delgado-Gomes, V., & Sanduleac, M. (2019). Smart Meters and Advanced Metering Infrastructure. Pathways to a Smarter Power System, 89–114.
[37] Rashed Mohassel, R., Fung, A., Mohammadi, F., & Raahemifar, K. (2014). A survey on Advanced Metering Infrastructure. International Journal of Electrical Power & Energy Systems, 63, 473–484.
[38] Pitì, A., Verticale, G., Rottondi, C., Capone, A., & Lo Schiavo, L. (2017). The Role of Smart Meters in Enabling Real-Time Energy Services for Households: The Italian Case. Energies, 10(2), 199.
[39] Vitiello, S.; Andreadou, N.; Ardelean, M.; & Fulli, G. (2022). Smart Metering Roll-Out in Europe: Where Do We Stand? Cost Benefit Analyses in the Clean Energy Package and Research Trends in the Green Deal. Energies, 15, 2340.
[40] ACER Annual Report on the Results of Monitoring the Internal Electricity and Natural Gas Markets in 2020—Energy Retail Markets and Consumer Protection Volume, November 2021, available online: http://acer.europa.eu/electricity/market-monitoring-report.
[41] Benchmarking Smart Metering Deployment in the EU-28, Final Report, March 2020, available online: https://op.europa.eu/en/publication-detail/-/publication/b397ef73-698f-11ea-b735-01aa75ed71a1/language-en.
[42] Gartner Glossary, Advanced metering infrastructure (AMI), available online: https://www.gartner.com/en/information-technology/glossary/advanced-metering-infrastructure-ami.
[43] Kaur, I. (2021). Chapter 29 - Metering architecture of smart grid. Advances in Nonlinear Dynamics and Chaos (ANDC). Design, Analysis, and Applications of Renewable Energy Systems. Academic Press, 687-704.
[44] Martins, J. F., Pronto, A. G., Delgado-Gomes, V., & Sanduleac, M. (2019). Smart Meters and Advanced Metering Infrastructure. Pathways to a Smarter Power System, 89–114.
[45] National Energy Technology Laboratory (NETL), Advanced Metering Infrastructure, December 2008, available online: https://netl.doe.gov/sites/default/files/Smartgrid/AMI-White-paper-final-021108--2--APPROVED_2008_02_12.pdf.
[46] Ghosal, A., & Conti, M. (2019). Key Management Systems for Smart Grid Advanced Metering Infrastructure: A Survey. IEEE Communications Surveys & Tutorials, 1–1.

[47] Koponen, P., Saco, L., Orchard, N., Vorisek, T.; Parsons, J., Rochas, C., Morch, A., Lopes, V., & Togeby, M. (2007). Definition of Smart Metering and Applications and Identification of Benefits. Deliverable D3 of the European Smart MeteringAlliance ESMA, State of art (pp. 5-7).
[48] Wang, Y., Chen, Q., Hong, T., & Kang, C. (2018). Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. IEEE Transactions on Smart Grid, 1–1.
[49] Colak, I., Sagiroglu, S., Fulli, G., Yesilbudak, M., & Covrig, C.-F. (2016). A survey on the critical issues in smart grid technologies. Renewable and Sustainable Energy Reviews, 54, 396–405.
[50] Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2013). A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges. IEEE Communications Surveys & Tutorials, 15(1), 5–20.
[51] Sauter, T., & Lobashov, M. (2011). End-to-End Communication Architecture for Smart Grids. IEEE Transactions on Industrial Electronics, 58(4), 1218–1228.
[52] Dusa, P., Novac, C., Purice, E., Dodun, O., & Slătineanu, L. (2015). Configuration a Meter Data Management System using Axiomatic Design. Procedia CIRP, 34, 174–179.
[53] Kabalci, Y. (2016). A survey on smart metering and smart grid communication. Renewable and Sustainable Energy Reviews, 57, 302–318.
[54] Wang, W., Xu, Y., & Khanna, M. (2011). A survey on the communication architectures in smart grid. Computer Networks, 55(15), 3604–3629.
[55] Taneja, M. (2013). Lightweight security protocols for smart metering. 2013 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia).
[56] Kuzlu, M., Pipattanasomporn, M., & Rahman, S. (2014). Communication network requirements for major smart grid applications in HAN, NAN and WAN. Computer Networks, 67, 74–88.
[57] Lunkeit, A., Vo, T., & Pohl, H. (2013). Threat modeling smart metering gateways. Proc. of European Conference on Smart Objects. Systems and Technologies (SmartSysTech), pp. 1–5.
[58] Otuoze, A. O., Mustafa, M. W., & Larik, R. M. (2018). Smart grids security challenges: Classification by sources of threats. Journal of Electrical Systems and Information Technology.
[59] Anzalchi, A., & Sarwat, A. (2015). A survey on security assessment of metering infrastructure in Smart Grid systems. SoutheastCon 2015.
[60] Ye Yan, Hu, R. Q., Das, S. K., Sharif, H., & Yi Qian. (2013). An efficient security protocol for advanced metering infrastructure in smart grid. IEEE Network, 27(4), 64–71.
[61] Murrill, B. J., Liu, E. C., & Thompson, R. M. (2012). Smart meter data: Privacy and cyber security. Congressional Research Service, Library of Congress, Tech. Rep.
[62] Anas, M., Javaid, N., Mahmood, A., Raza, S. M., Qasim, U., & Khan, Z. A. (2012). Minimizing Electricity Theft Using Smart Meters in AMI. 2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing.
[63] Gartner Glossary, Load forecasting, available online: https://www.gartner.com/en/information-technology/glossary/load-forecasting#:~:text=Load%20forecasting%20minimizes%20utility%20risk,and%20renewable%20generation%20predictive%20modeling.
[64] Rahman, A., Srikumar, V., & Smith, A. D. (2018). Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. Applied Energy, 212, 372–385.
[65] Namlı, E., Erdal, H., & Erdal, H. I. (2018). Artificial Intelligence-Based Prediction Models for Energy Performance of Residential Buildings. Environmental Science and Engineering, 141–149.
[66] Hong, T., & Fan, S. (2016). Probabilistic electric load forecasting: A tutorial review. International Journal of Forecasting, 32(3), 914–938.
[67] Khatoon, S., Ibraheem, Singh, A. K., & Priti. (2014). Effects of various factors on electric load forecasting: An overview. 2014 6th IEEE Power India International Conference (PIICON).
[68] Kyriakides, E., & Polycarpou, M. (n.d.). Short Term Electric Load Forecasting: A Tutorial. Trends in Neural Computation, 391–418.
[69] Gajowniczek, K., & Ząbkowski, T. (2014). Short Term Electricity Forecasting Using Individual Smart Meter Data. Procedia Computer Science, 35, 589–597.
[70] Tucker, N., Moradipari, A., & Alizadeh, M. (2020). Constrained Thompson Sampling for Real-Time Electricity Pricing with Grid Reliability Constraints. IEEE Transactions on Smart Grid, 1–1.
[71] Soudan, B., & Darya, A. (2020). Autonomous Smart Switching Control for Off-grid Hybrid PV/Battery/Diesel Power System. Energy, 118567.
[72] Papalexopoulos, A. D., & Hesterberg, T. C. (n.d.). A regression-based approach to short-term system load forecasting. Conference Papers Power Industry Computer Application Conference.
[73] Henselmeyer, S., & Grzegorzek, M. (2021). Short-Term Load Forecasting Using an Attended Sequential Encoder-Stacked Decoder Model with Online Training. Applied Sciences, 11, 4927.

[74] Christiaanse, W. (1971). Short-Term Load Forecasting Using General Exponential Smoothing. IEEE Transactions on Power Apparatus and Systems, PAS-90(2), 900–911.
[75] Liu, K., Subbarayan, S., Shoults, R. R., Manry, M. T., Kwan, C., Lewis, F. I., & Naccarino, J. (1996). Comparison of very short-term load forecasting techniques. IEEE Transactions on Power Systems, 11(2), 877–882.
[76] Cheng, D., Xu, J. & Zheng, Z. (2017). Analysis of Short-term Load Forecasting Problem of Power System Based on Time Series. Autom, vol. 11, pp. 99–101.
[77] Zhang, F. & Zhang, F. (2017). Power Load Forecasting in the Time Series Analysis Method Based on Lifting Wavelet. Electr. Autom, vol. 39, pp. 72–76.
[78] Alberg, D., & Last, M. (2018). Short-term load forecasting in smart meters with sliding window-based ARIMA algorithms. Vietnam Journal of Computer Science.
[79] Chakhchoukh, Y., Panciatici, P., & Bondon, P. (2009). Robust estimation of SARIMA models: Application to short-term load forecasting. 2009 IEEE/SP 15th Workshop on Statistical Signal Processing.
[80] Bercu, S., & Proïa, F. (2013). A SARIMAX coupled modelling applied to individual load curves intraday forecasting. Journal of Applied Statistics, 40(6), 1333–1348.
[81] Ghofrani, M., Hassanzadeh, M., Etezadi-Amoli, M., & Fadali, M. S. (2011). Smart meter based short-term load forecasting for residential customers. 2011 North American Power Symposium.
[82] Liu, N., Tang, Q., Zhang, J., Fan, W., & Liu, J. (2014). A hybrid forecasting model with parameter optimization for short-term load forecasting of micro-grids. Applied Energy, 129, 336–345.
[83] Sun, Y., & Zhang, Z. (2005). Short-Term Load Forecasting Based on Recurrent Neural Network Using Ant Colony Optimization Algorithm. Power Syst. Technol, vol 29, pp. 59–63.
[84] Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2017). Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network. IEEE Transactions on Smart Grid, 1–1.
[85] Gers, F. A. (1999). Learning to forget continual prediction with LSTM. 9th International Conference on Artificial Neural Networks: ICANN '99.
[86] Afrasiabi, M., Mohammadi, M., Rastegar, M., Stankovic, L., Afrasiabi, S., & Khazaei, M. (2020). Deep-Based Conditional Probability Density Function Forecasting of Residential Loads. IEEE Transactions on Smart Grid, 1–1.
[87] Gerossier, A., Girard, R., Bocquet, A., & Kariniotakis, G. (2018). Robust Day-Ahead Forecasting of Household Electricity Demand and Operational Challenges. Energies, 11(12), 3503.
[88] Wang, Y., Zhang, N., & Chen, X. (2021). A Short-Term Residential Load Forecasting Model Based on LSTM Recurrent Neural Network Considering Weather Features. Energies, 14(10), 2737.
[89] Wu, R., & Bao, Z. (2018). Research on Short-term Load Forecasting Method of Power Grid Based on Deep Learning. Mod. Electr. Power, vol. 35, pp. 43–48.
[90] Rahman, A., Srikumar, V., & Smith, A.D. (2018). Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. Appl. Energy, vol. 212, pp. 372–385.
[91] Marino, D.L., Amarasinghe, K., & Manic, M. (2016). Building Energy Load Forecasting using Deep Neural. IECON.
[92] Janacek, G. (2009). Time Series Analysis Forecasting and Control. Journal of Time Series Analysis.
[93] Montgomery, D. C., Jennings, C. L. & Kulahci M. (2015). Time Series Analysis and Forecasting, vol.2.
[94] Chatfield, C. (2001). Time-Series Forecasting. Basics of Time-Series Analysis, 131–155.
[95] Box, G. E. P., Jenkins, G. M. & Reinsel, G. C. (2008). Time Series Analysis.
[96] Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena, 132306.
[97] Bengio, Y., Frasconi, P., & Simard, P. (n.d.). The problem of learning long-term dependencies in recurrent networks. IEEE International Conference on Neural Networks.
[98] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.
[99] https://data.4tu.nl/articles/dataset/Energy_consumption_data_of_the_GridFlex_Heeten_project/14447257/1?file=27671892
[100] https://www.dask.org/
[101] Wang, Z., Li, J., Zhu, S., Zhao, J., Deng, S., Zhong, S., … Gan, Z. (2019). A Review of Load Forecasting of the Distributed Energy System. IOP Conference Series: Earth and Environmental Science, 237, 042019.
[102] Wang, Y., Zhang, N., & Chen, X. (2021). A Short-Term Residential Load Forecasting Model Based on LSTM Recurrent Neural Network Considering Weather Features. Energies, 14(10), 2737.
[103] Abadi, M., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from https://www.tensorflow.org/.

[104] Kingma, D. P. & Ba, J. (2014). Adam: A Method for Stochastic Optimization. 3rd International Conference for Learning Representations, San Diego.
[105] Fan, G.-F., Guo, Y.-H., Zheng, J.-M., & Hong, W.-C. (2019). Application of the Weighted K-Nearest Neighbor Algorithm for Short-Term Load Forecasting. Energies, 12(5), 916.