Vaasan yliopisto
UNIVERSITY OF VAASA

Fatama Sultana Liza

Customer Energy Flexibility Forecasting with Different Machine Learning Models

School of Technology and Innovations
Master's thesis in Smart Energy Programme

Vaasa 2024

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Fatama Sultana Liza
Title of the thesis: Customer Energy Flexibility Forecasting with Different Machine Learning Models
Degree: Master of Science in Technology
Discipline: Smart Energy
Supervisor: Hannu Laaksonen
Co-supervisor: Hosna khajeh
Year: 2024 **Pages:** 122

ABSTRACT :

Most concerns about the sustainability of modern society are centred around energy issues. To address these, the power system is transitioning towards a more intelligent, flexible, and interactive system with higher penetration of renewable energy generation. A key enabler of this transition is energy flexibility, which focuses on the demand response of consumers at different times. It is the traceability of the energy flow among individual energy consumers. This master's thesis focuses on the demand response-based flexibility of the consumers at different timescales. Accurate load forecasting of individual customers is increasingly vital for future grid planning and operation. The main objective of this thesis was to identify the most flexible consumer among seven consumers and to identify the most effective machine learning (ML) model for forecasting energy flexibility across time horizons and time scales. This study investigates how different machine learning models—such as recurrent neural networks (RNN), gradient boosting, linear regression, and long short-term memory (LSTM)—can be used in the energy forecasting process. This study utilised a total of 11 distinct machine learning models. The experimental findings clearly demonstrate that the proposed model is superior to the others in terms of accuracy in predicting consumption, as measured by the root mean square error (RMSE).

In addition, the model also evaluates the degree of flexibility of these households to adjust or reduce energy consumption in response to price fluctuations. This study proposes the implementation of a three-data model strategy to effectively manage load flexibility forecasting (Customer profit maximization). This technique aims to provide flexibility services, like congestion management, peak shaving to the local distribution system operator (DSO) and integrate renewable energy sources by leveraging several features like advance forecasting and analytics, demand response.

KEYWORDS: Machine Learning, Energy flexibility, Demand response, Forecasting.

Acknowledgments

This master’s thesis was conducted at the University of Vaasa in the frame of the Smart Energy Programme. I wish to express my sincere gratitude to all the people who have supported me during realization of this thesis.

In the first instance, it is my pleasure to thank my thesis supervisor, **Professor Hannu Laaksonen,** for the opportunity to be associated with him. His suggestions and encouragement are irreplaceable; hence, I would not have been able to finish this thesis on my own. I would also like to thank my co-supervisor, **Hosna Khajeh,** for her endless support as well as her remarks on the work and her suggestions in this work.

I also want to accord my gratitude to the University of Vaasa´s Smart Grid 2.0 project, which voluntarily offered their data for this research. These were essential for the achievement of this research and they played a huge role.

Moreover, it is important to express my gratitude to the members of my family and friends for their contributions to completing this thesis and conducting all the research. They helped me make sense of whatever I have been through and to endure it. I want to dedicate this thesis to my parents, **Firaoj Alam** and **Shahanara Begum.** They are the reason I have achieved this milestone and their sweet words inspired me when I was down.

Finally, the successful outcomes of this work can be attributed to the support from the Smart Grid 2.0 – Next Generation Distribution Grid project, implemented within the framework of Business Finland.

Contents

Figures

Tables

Abbreviations

ML	Machine Learning
AI	Artificial Intelligence
RES	Renewable Energy Sources
EMS	Energy Management Systems
DSM	Demand-Side Management
GHS	Green House gas
DSF	Demand-Side Flexibility
ADR	Automated Demand Response
DES	Distributed Energy System
EDA	Exploratory Data Analysis
AC	Autocorrelation
DLC	Direct Load Control
LR	Linear Regression
EL	Ensemble Learning
GBR	Gradient Boosting Regressor
RNN	Recurrent Neural Networks
NNA	Neural Network Architecture
GRU	Gated Recurrent Units
LSTM	Long short-term memory
kW	Kilowatts
RMSE	Root Mean Squared Error
LEC	Local Energy Community
ReLU	Rectified Linear Unit
RTP	Real Time Price
CPP	Critical Peak Price
TOU	Time-Of-Use
DSO	Distribution System Operator

1 Introduction

The transition towards renewable energy sources in the constantly changing landscape of energy systems emphasises the urgent need for flexible and responsive energy management techniques. An effective method to handle fluctuations in energy supply is through the utilisation of energy flexibility. Energy flexibility, defined as the ability of customers to adjust their energy consumption or production patterns, plays a crucial role in balancing supply and demand, optimizing grid operations, and enhancing grid resilience. The significant role for this energy flexibility is towards a transition to a sustainable and resilient energy system and integrating renewable energy sources, such as wind and solar, is a critical component of this transformation. Moreover, the intermittent nature of renewable resources poses challenges for grid stability and reliability. In most countries, power supply still mainly relies on the combustion of traditional fossil fuels which has brought serious environmental problems (Wen et al.,2019). At the same time, electricity consumption has increased significantly with the explosive growth of population and the rapid development of economy, especially for the residential electricity consumption which shows an increasing trend. Now a days, to increase the grid stability and reliability, the modern smart grid is to promote demand and supply interaction and shifting the major position of the customers inwards the energy distribution systems and real-time operations. The advantages of this flexibility are that, a large number of customers are suitable to forecast, monitor and manage their 24- h energy profiles, which can render them to some extent advantages in terms of consumption reductions and financial benefits (Ahmad et al.,2020)

In smart grids, the adoption of Renewable Energy Sources (RES) involves the use of several applications and essential component systems such as Energy Management Systems (EMS), distribution management systems, outage management systems, big data analytics, and load forecasting (Lee et al., 2024). Moghimi et al. (2024) contend that conventional energy management systems (EMS) require a more efficient optimisation technique to effectively reduce energy consumption and costs.

On the contrary, the significance of aggregated customers and residential users in EMS is growing because of the increasing accessibility of energy generation (Meng et al., 2024). According to researchers, Meng et al. (2024) have found that home clients are crucial for actively participating in this field. In addition, the utility company has also implemented several strategies to enhance the apparent simplicity of pricing conditions. Provide clients with up-to-date information on the existing electricity pricing through real-time updates. This assists clients in making timely and well-informed decisions. Moreover, effectively utilize the renewable energy renewable energy resources into power systems increases the need for flexibility services and also system operators need to utilize more flexible energy resources from all levels of the system in order to fulfil flexibility needs (Firoozi et al.,2021).

Huotari et al. (2024) reported that households in Europe, North America, and Asia consume 20–30% of the total energy used in those regions. Additionally, worldwide energy consumption has consistently risen in the past few decades. Hence, in order to address the increasing demand for energy and promote efficient energy consumption, the rising significance of Artificial Intelligence (AI) and Machine Learning (ML) methods are becoming acknowledged as a way to create inventive strategies that enhance energy utilisation for consumers.

To developed a reliable and robust algorithm for accurate energy demand prediction is indispensable and also challenging because the performance of a forecasting algorithm may be affected by various factors, such as data quality, geographic diversity, forecast horizon, customer segmentation and the forecast origin (Ahmad et al.,2020).

The main objectives of the research are that use different types of machines learning models to estimate which forecasting model will be accurate for predict future energy consumption patterns and flexible customer based on different types of features.

1.1 Background and motivation

According to Hardmeier et al. (2024), European electricity market is mainly dependent on the energy-only market. Customers always pay a fixed rate, meaning that the price per kilowatt hour remains constant regardless of the time of usage. Alternatively, users may choose a day-and-night tariff, which offers a slightly cheaper price during the night compared to the day. Furthermore, due to price consistency, consumers are unaware of the actual supply and demand dynamics in the electricity market. However, as electrification expands and the proportion of new renewable energy sources such as solar and wind power increases, the need for flexibility in maintaining a dependable energy system is growing. An adaptable energy system can provide uninterrupted operation even in the face of abrupt and significant fluctuations in energy supply. In order to achieve flexibility at the system level, it is crucial to regard demand-side flexibility as an essential element.

The integration of renewable energy sources, such as solar and wind power, into the electrical system results in a notable increase in unpredictability and fluctuation in the energy supply. In addition, the integration of system stability is currently facing unprecedented problems due to the intermittent nature of renewable energy generation and the complex dynamics of utility-customer interactions (Kong et al., 2017). According to the European Union (EU), urban buildings are responsible for 40% of global energy consumption and 33% of greenhouse gas (GHG) emissions (Moghimi et al., 2024).
To mitigate energy consumption and GHG, one of the most important challenges to this endeavour lies in identifying the most flexible consumer clusters with similar consumption behaviours (Michalakopoulos et al., 2024). In terms of electricity consumption, the residential sector's contribution is approximately 29% compared to transport, agricultural and forestry, services, and industry. The disaggregated nature of the assets (households) and their relatively small size, in terms of providing a significant proposition to the market, have posed some challenges (Lucas et al., 2019).

To this end, accurate electric load forecasting at the level of residential customers can significantly facilitate power system operations. As a result, there is an increasing reliance on demand-side management (DSM) strategies to maintain grid stability. From the utilities' point of view, researchers Kong et al. (2017) also mentioned that the accurate load forecasts for individual customers are available, these electricity suppliers can rely on such information to target the best groups of customers with the highest potential to participate in DR programmes in the event of power deficiency.

Energy flexibility, an essential aspect of smart grids, involves the adaptation of energy use to align with the requirements of the system. Customers within distribution networks have the ability to offer this adaptability by using controlled appliances that can respond to signals such as changes in electricity pricing. System operators and aggregators must predict the potential for regional flexibility by analysing past behaviours in order to ensure efficient activation of grid services. To optimise flexibility services across different periods, accurate estimates must consider the varying activation dates and durations.

In the field of energy flexibility, demand response (DR) is a promising method for balancing supply and demand in power systems, with a high share of variable renewable energy generation offering flexibility to the market (Lucas et al., 2019). Michalakopoulos et al. (2024) specified that during periods of high demand or grid instability, users adjust their electricity consumption by reducing or shifting it, alleviating pressure on the grid and mitigating the risk of blackouts. DR enhances both grid dependability and environmental sustainability by aligning energy consumption with the availability of renewable energy, thereby minimising reliance on backup power sources. In addition, DR offers chances for cost reduction, as consumers who participate can take advantage of incentives and enjoy lower electricity rates during off-peak hours.
This makes a significant contribution to improving efficiency, saving power consumption, enhancing dependability, and conserving power for decentralised frameworks within a smart grid model (Meng et al., 2024).

1.2 Problem statement

Although the significance of energy flexibility is acknowledged, there are considerable obstacles in accurately forecasting how aggregated customer groups can collectively modify their energy consumption. These issues arise from the variations in consumption patterns, the design of building structures (which account for 20% to 40% of total energy usage), the differences in how different consumer groups respond, and the complications presented by varied time scales and horizons in energy forecasting. Precise prediction of energy flexibility across various time intervals (ranging from hourly modifications to seasonal changes) and timeframes (from rapid reactions to long-term strategizing) is crucial for efficient grid administration and the execution of adaptable energy strategies. Smart grids are increasingly valuing the inclusion of both customer and residential users in order to accommodate the growing availability of power production.

1.3 Research objectives

This thesis aims to use a different machine learning (ML) comprehensive forecasting model that:

- To assess the energy consumption data of seven different house in Vaasa at different time scales (e.g. daily, weekly, monthly and holidays).
- Differentiates the impacts of various time scales and horizons on energy flexibility.
- Prepare the data and identify the critical features for forecasting energy flexibility (feature selection).
- To run different types of forecasting models based on different features such as air temperature, month, week, hour, holydays and day-ahead price and so on, predicts the future energy consumption pattern for different customers.

- Develop different forecasting models, including regression-based, ensemble-based and neural network-based models.
- Analyze the machine learning model's performance by analyzing its accuracy and efficiency with an evaluation matrix and different co-relation coefficient such as Pearson and spearman
- Accurately predicts the energy flexibility of different customers.

1.4 Thesis outlines

The study is organised as follows: in chapter 2, it highlights the significance of forecasting and demand response in the energy sector. Chapter 3 presents a comprehensive introduction of several machine learning models with mathematical explanation. Chapter 4 primarily focuses on the utilisation of several correlation matrices inside the model to determine the correctness and relationships among distinct characteristics. Chapter 5 primarily depicts the most adaptable consumer based on several factors. Chapter 6 elucidates the technique employed in this study. Eventually, chapter 7 and 8 present the results of this research.

1.5 Significance of the study

The results of this study will offer important insight on the adaptable consumer's ability to adjust their energy usage based on various factors such as price, time scale, past behaviour of customers and so on. This also improves the prediction ability of energy managers and policy-makers. This effort intends to contribute to more reliable and efficient energy systems, especially in areas where renewable energy sources are widely used, by increasing the accuracy of flexibility forecasts.

2 Forecasting and demand response in energy sector

The significance of forecasting and demand response in the energy sector is increasing due to its ability to facilitate efficient management and optimisation of energy resources. The utility companies may enhance their generation and distribution techniques and improve resource allocation more effectively by making accurate projections about future energy demand, leading to reduced costs and increased system dependability (Ahmad et al., 2020). Furthermore, the important elements of the energy industry are demand response and energy forecasting. Demand response (DR) refers to the capacity of customers to modify their energy use in reaction to price signals or grid circumstances. On the other hand, energy forecasting is the process of projecting future energy demand and supply to optimise the generation, distribution, and use of power. These two features are intricately interconnected and frequently coincide (Hong et al., 2020).

In addition, the energy industry may efficiently manage and enhance power use by integrating demand response and energy forecasting. An effective approach to doing this is by employing forecasting models and techniques, such as machine learning-based approaches and stochastic methods, to anticipate whether a certain appliance will be turned on or off in the near future. This may be helpful in the strategic development and execution of demand response initiatives, where customers are motivated to curtail their power use during periods of high demand or move it to times of lower demand (Albadi & El-Saadany, 2008). One method used to achieve this integration is through the use of demand management techniques, which involve forecasting and planning how demand for goods and services will be met and actively seeking ways to reduce or moderate that demand (Vardakas et al., 2015). The energy sector may achieve more sustainable and efficient operations by using both energy forecasting and demand response. An essential facet of combining demand response with energy forecasting is assessing the efficiency of energy-saving initiatives, specifically in regards to appliance load forecasting for demand response. In order to determine the efficiency of energy-saving programmes for demand response, it is essential to examine the precision of appliance load forecasts.

A frequently employed strategy involves using machine learning and stochastic techniques to forecast the activation or deactivation of a certain appliance in the immediate future. Aside from predicting appliance consumption, it is crucial to measure the energy efficiency and cost savings that come with using energy-efficient equipment. The energy industry is experiencing tremendous advancements in renewable energy generation, which in turn necessitates a growing need for flexibility in the energy system (McKenna et al., 2021).

Currently, the demand for electricity is becoming more dynamic, with a larger proportion of intermittent renewable power generation from sources such as solar photovoltaics and wind turbines. As a result, the power grid is confronting a rising problem in effectively managing the real-time balance between supply and demand (Li et al., 2021). Alanne & Sierla (2022) mentioned, the transition to sustainable urban development requires energy systems to be modified to handle the increasing incorporation of variable renewable energy sources in a setting with stringent demands for resilience, flexibility, and energy efficiency. Nevertheless, buildings must possess the capacity to adjust to varying external factors, such as the demands of the users, changing climate conditions, and shifting grid pricing. The role of customers has undergone substantial transformation in the past two decades due to the reorganisation of the electrical system. Generally, residential consumers do not observe short-term variations in prices. However, most individuals are charged a fixed rate for each unit of power they use. To fully utilise the potential of demand-side flexibility (DSF) in the residential sector, consumers must be familiar with the temporal variations in power costs observed on wholesale markets (McKenna et al., 2021). In order to meet flexibility requirements, future system operators must make use of a wider range of adaptable energy resources from all levels of the system. Furthermore, a group of consumers that have been combined into a local energy community (LEC) have the ability to serve as valuable resources that can contribute to meeting a portion of the necessary flexibility. Precise prediction of the adaptable capabilities of a local exchange carrier (LEC) is crucial in this context (Firoozi et al., 2021).

Furthermore, the incorporation of demand-side flexibility into the future energy system and its inclusion in system-level models are crucial for achieving cost savings. One effective approach to improving the sustainability of energy systems is to utilise demand-side management (DSM) or demand-side flexibility (DSF) in residential settings. This involves applying ways to intelligently regulate and modify energy consumption patterns in residential dwellings. Households may synchronise their energy use with periods of ample renewable energy supply, such as when solar or wind power output is at its highest, by optimising the timing and scale of their energy usage. This strategy optimises the utilisation of renewable energy sources while enhancing the overall efficiency and robustness of the energy infrastructure (McKenna et al., 2021).

This thesis explores the price-based demand response and flexibility of various households using different machine learning models. The primary objective is to determine which house exhibits greater flexibility across several characteristics. In addition, analysis of real-time consumer data reveals that more customer flexibility under real-time energy tariffs leads to increased grid stability, as seen by a decrease in the average peak-to-average ratio with the implementation of real-time pricing (Cortez et al., 2024). On the one hand, Vardakas et al. (2014) specified that, highly flexible consumers showing more responsiveness in demand-side management (DSM) or demand-side flexibility (DSF) programmes can greatly contribute to the stability and resilience of the electrical grid.

2.1 Energy flexibility: definition and concepts

The term energy flexibility describes a system's or organization's capacity to modify its production or consumption of energy in response to shifts in pricing, supply, or demand. The ability to adapt is essential for attaining a durable and effective energy system (Lund et al., 2015). It enables the efficient utilisation of energy, minimization of waste, and incorporation of intermittent renewable sources into the power system. Energy flexibility allows a system or organisation to adjust its energy usage and generation in response to factors like demand, supply, and cost (Jakhar, 2017).

According to Li et al. (2021), Energy flexibility refers to the capacity of an energy system or part to adapt its operation in order to optimise performance, improve dependability, and facilitate the integration of renewable energy sources. This adaptation is done in response to changing conditions, needs, or external signals. The concepts associated with energy flexibility encompass demand-side management (DSM), demand response (DR), load shifting, energy storage, grid-interactive buildings, virtual power plants (VPPs), and predictive analytics, all of which are highly pivotal for achieving a high level of energy flexibility.

Moreover, in predictive analytics, we mainly focus on the different machine learning models to find the most flexible house in our data set. This thesis paper primarily examines the concept of building energy flexibility, which refers to a building's ability to adjust its energy demand and generation in order to support the electric grid based on local ambient conditions, such as weather. It emphasises the importance of maintaining the needs of building occupants, such as thermal comfort and productivity, while implementing these adjustments. Figure 1 illustrates the fundamental performance criteria for building energy flexibility (Li et al., 2021).

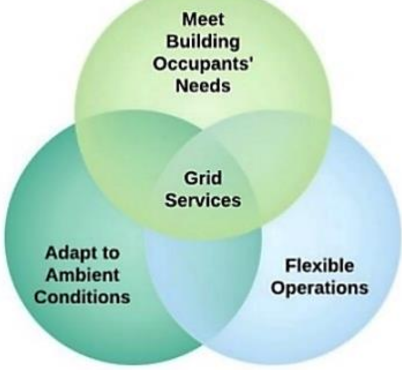

Figure 1. Concept of triple bottom line in building energy flexibility (Li et al., 2021)

Energy flexibility is essential for the effective and dependable functioning of energy systems. It is also critical for successfully integrating renewable energy sources and achieving a more sustainable energy future.

2.2 Demand response programs and flexibility

Demand response (DR) program and flexibility are essential components of the modern energy landscape. DR is a strategy in the energy industry that includes controlling electricity usage by adjusting the amount of power used by consumers based on signals from the grid operator. Essentially, it involves modifying the energy consumption patterns in residential, commercial, or other institutions to match the current supply and demand conditions on the power grid. The conventional balance between grid supply and demand is mostly preserved by the modulation of power generation rates, activation of backup generation, or the importation of electricity from interconnected utilities. Buildings equipped with adaptable operating capabilities have the ability to adjust their power use in response to signals from the electrical grid or changes in price. DR is the term used to describe the action of adjusting power consumption on the demand side (Niu et al., 2020).

According to Niu et al. (2020) mentioned that, DR program are important drivers of energy flexibility. Traditionally, DR has focused on large commercial and industrial customers. But as the technology matures, more and more small commercial and residential customers are getting involved. According to a recent study by Automated demand response (ADR), a concept started in the early 2000s, aims to facilitate timely and predictable responses for system operators and flexible customer participation. These programmes can be broadly categorized into incentive-based and price-based program. Generally, demand response (DR) programs have a great potential to unlock energy flexibility in a distributed energy system (DES) by implementing an implicit mode, also known as a price-driven mode, such as a time-of-use (TOU) tariff, real-time price (RTP) and critical-peak price (CPP). On the other hand, in an explicit mode, allowing a DES to participate the regulation of the power interacted with the main power grid.

A price-based DRP leads to substantial modifications in the power consumption patterns in response to the market price variations. DR can be described more accurately as modifying the power consumption with respect to the usual consumption, in response to the market price or incentive to motivate the consumer to change the consumption pattern (Mansouri et al., 2021). It has been noticed that time-of-use pricing is the most used mechanism worldwide for the time of consumption of electricity. In this system, consumers would react to price signals and reduce their bill. This mechanism divides the 24 hours of the day into three or four periods: peak, off-peak, and valley periods.

Each of these periods is associated with a fixed price. These prices may vary for different hours of the day, different days of the week, or seasons of the year. The differences in the prices are the incentive for consumers to reduce their consumption or shift their leads to other periods. These programs are mandatory and arbitrary programs. Consumers are able to participate in arbitrary programs and give up after the agreed period.

Similarly, Real-time pricing is another price-based DRP, with hourly-varying pricing. The type of this DR program is arbitrary. Once the consumers enter this program, they should continue with the contract for a given period. The more substantial the variations of the market prices, the more the load shifting of consumers will be. Another one is critical peak pricing is a combination of the TOU and RTP mechanisms. CPP is associated with a predetermined high price designed by distribution companies to apply over peak intervals. These tariffs are called for a limited number of days or hours of the day with relatively short cautions. The consumers will receive a price discount over off-peak hours through this mechanism.

On the other hand, incentive-based programmes are planned by distribution companies, service-providing entities, and local system operators with respect to the price consideration and the specific features of generators and the system. These DR programs include direct load control (DLC), interruptible/curtailable services, demand bidding/buyback programs, emergency DRPs, capacity market programmes, and ancillary service market programs (Mansouri et al., 2021).

The power systems and infrastructure are undergoing an imminent transition to describe the developed variability in production and actual net energy demand, introduced by a series of power generators, specifically solar and wind. Several methods to decrease the adverse ramping impacts have been proposed, i.e., demand response, resources, and total energy prediction, increased storage capacities, etc. The essence of complete solutions for combining the higher numbers of parameters in solar and wind production is to develop the available flexibility options in the smart grid (Ahmad et al., 2020).

Mansouri et al. (2021), DR more accurately as modifying power consumption with respect to usual consumption, in response to the market price or an incentive to motivate the consumer to change the consumption pattern. According to Alanne & Sierla (2022), the advancement of sustainable buildings, communities, and societies relies on the energy systems' capacity to handle the growing integration of intermittent renewable energy sources, while meeting the requirements of energy efficiency, flexibility, and resilience.

On the other hand, buildings must possess the flexibility to adjust to varying external factors such as user requirements, shifting climatic patterns, and variable electricity prices. In order to predict DR, several machine learning techniques have been employed for demand response in seven residential buildings. Also, determine the level of flexibility in each house by analysing numerous factors such as price, temperature, and time. Furthermore, DR program also play a pivotal role in the electricity market; they balance supply and demand by taking advantage of load flexibility. Demand response (DR) is one of the demand side management (DSM) methods; it was introduced with the aim of mitigating system reliability problems and price spikes.

3 Machine learning modelling techniques

Machine learning (ML) is a discipline that centres on empowering computers with the capacity to acquire knowledge and improve performance without the need for explicit programming. This process encompasses acquiring novel information, improving abilities through repeated application, structuring knowledge in practical manners, and discovering new insights through observation and experimentation. The primary objective is to build computer skills that closely resemble human learning, which is a significant and continuous challenge in the field of artificial intelligence (AI) (Carbonell et al., 1983).

Nasteski. (2017) mentioned, ML is a multidisciplinary area that overlaps with other fields such as information technology, statistics, artificial intelligence, and neuroscience. The focus is on creating algorithms that let computers to recognise patterns and consistencies in data, thus acquiring knowledge from it. These algorithms are specifically developed to mimic human learning processes, adjusting to different levels of intricacy and surroundings. This sophisticated computational methods to process complex data in several dimensions for a wide range of applications, marking a significant advancement from its original capabilities.

3.1 Linear regression technique

ML is widely employed across several disciplines to address complex issues that are not easily solvable using traditional computer methods. Linear regression (LR) is a basic and widely used machine learning technique. It is a mathematical methodology employed for doing predictive analysis. LR enables the projection of continuous or mathematical variables. The concept of linear regression was initially proposed by Sir Francis Galton in 1894. LR is a mathematical technique used to assess and measure the connection between variables under consideration (Maulud et al., 2020). LR is a technique employed to represent the connection between a dependent variable and one or more independent variables by fitting a linear equation to observed data.

3.1.1 Linear Regression

Linear regression (LR) is a prevalent and extensively utilised modelling approach in the fields of statistics and machine learning. The objective is to build a direct correlation between the input variables and the target variable. The model postulates a linear combination of the input characteristics in order to forecast the continuous output variable. The input variables' coefficients are calculated by the utilisation of several optimisation techniques, such as the method of least squares. LR is easily implementable and interpretable, making it a favourable option for situations where there are linear associations between variables (Zhou et al.,2023).

3.1.1.1 Formulation and interpretation

The linear regression model presumes that the dependent variable (Y) may be represented as a linear combination of an independent variable (X), together with an error term (ε) that accommodates for deviations.
The model is defined by the equation 1 (Panel., 2014).

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \tag{1}$$

Where β_0 is the intercept, β_1 the slope, and εi the error term for each observation i.
The linear regression model presumes a linear correlation between the variables, where the dependent variable can be forecasted based on the impact of the independent variable.
The error term ε plays a crucial role in linear regression. It captures the discrepancy between the actual observed values (Y) and the predicted values ($\widehat{Y}$). These predicted values are calculated by using the model's equation 2 (Panel., 2014).

$$\widehat{Y} = \beta_0 + \beta_1 X. \tag{2}$$

The error term ε is derived from the difference between the observed values (Y) and those predicted by the linear model ($\widehat{Y}$).

$$\varepsilon_i = Y_i - \widehat{Y}_i$$

Here, ε_i represents the error term for the i th observation, Y_i is the actual observed value and $\widehat{Y}_i$ is the predicted value for the i th observation, calculated from the linear regression model. This disparity encompasses the unpredictability and unaccounted for fluctuations in the dependent variable, Y, that are not explained by the independent variable, X. It recognises the intricate nature of real-world data, in which several unidentified factors influence Y. By incorporating the error term, one can evaluate the precision of the model's predictions and get valuable insights into how the model could be enhanced by including alternative independent variables. The error term can be understood as the cumulative impact of variables that are not accounted for in the model (Panel., 2014).

3.1.2 Poly linear regression

Poly-linear regression is a reliable statistical technique that enables the modelling of intricate interactions between variables. This approach enhances simple linear regression by integrating polynomial terms, hence enhancing the model's adaptability and enhancing its capacity to accurately represent the data.

Polynomial Regression is an extension of linear regression that models the relationship between dependent and independent variables using a polynomial function. This methodology allows the model to effectively capture complex and non-linear connections within the dataset, offering a more adaptable framework in contrast to basic linear regression. Non-linearity in data is the occurrence when the connection between independent factors and a dependent variable cannot be precisely stated using a straight line. However, many relationships display curves or other intricate patterns, suggesting that the impact of the independent factors on the dependent variable varies at various rates over the data range. Polynomial Regression is classified as a specific instance of linear regression because it is linear in the coefficients that are derived from the data, even though it models a nonlinear connection between variables (Panel., 2014).

According to Stulp & Sigaud (2015), poly-linear regression might alleviate some constraints linked to conventional linear regression models, such as incomprehensible estimates and hypothesis testing. Poly-linear regression enhances prediction accuracy by including polynomial terms and interactions between variables, enabling a more comprehensive representation of the complexity inherent in real-life scenarios.

3.1.2.1 Formulation and interpretation

The general form of the Polynomial Regression model can be expressed in equation 3 (Panel., 2014).

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_n X^n + \varepsilon, \tag{3}$$

Here:

• Y represents the dependent variable.

• X is the independent variable.

• $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients of the model, indicating the influence of each polynomial term of X on Y.

• Xn represents the nth polynomial term of X, allowing the model to capture nonlinear relationships.

• ε is the error term, accounting for the variability in Y not explained by the polynomial terms of X.

This formulation enables the Polynomial Regression model to fit more complex curves than a simple linear regression model, providing a better approximation of the underlying relationship when the data exhibits nonlinearity (Panel., 2014).

3.2 Ensemble learning

Ensemble learning is a potent methodology in machine learning that entails generating and merging many models to enhance prediction performance. Ensemble learning utilises the advantages of several algorithms to achieve more precise and resilient predictions. This technique is founded on the notion that any individual model may include its own constraints and prejudices. However, by amalgamating them, their inaccuracies may be offset, leading to a more precise and dependable total forecast (Livieris et al., 2020). Ensemble learning allows us to harness the variety and combined intelligence of numerous models to surpass the constraints of individual models and attain superior prediction performance. Ensemble learning is a method in machine learning that integrates different models to enhance predicted performance (Brownlee, 2021).

Moreover, ensemble approaches enhance the overall accuracy and dependability of machine learning solutions by training many models and merging their predictions showed in figure 2. (Cunningham, 2022).

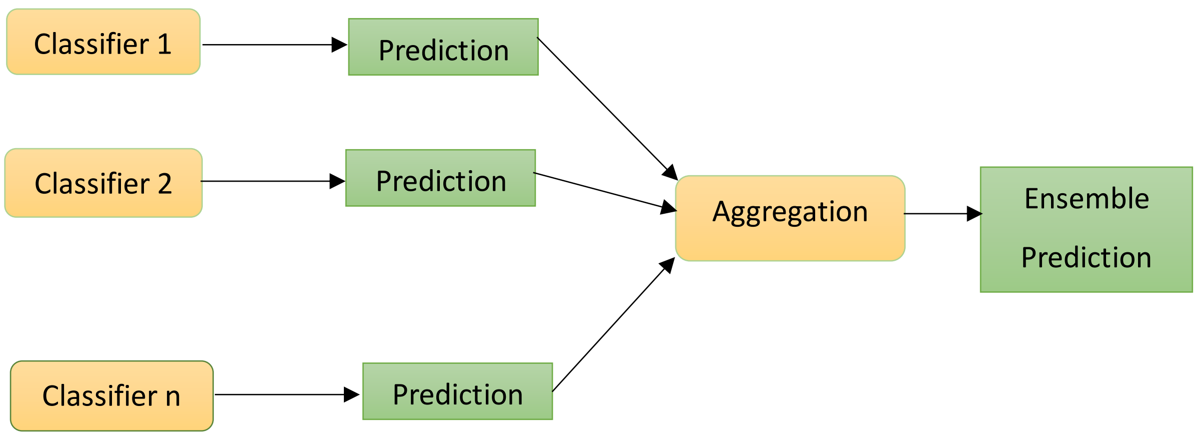

Figure 2. Ensemble model structure (Cunningham, 2022)

3.2.1 Gradient Boosting Regressor (GBR)

Gradient-boosting regression (GBR) is a powerful machine learning algorithm, particularly well-suited for regression tasks. It has been an ensemble method made by adding weak leaners, typically decision trees, to enhance the predictions. GBR’s ability to grasp intricate relationships between features and the target variable, coupled with its robustness in handling large datasets even in the presence of noise and outliers, makes it a valuable tool for predictive modelling (Houda et al.,2022).

3.2.1.1 Formulation and interpretation

GBR is based on minimising the loss function, which determines the difference between predictions and actual target values. Every subsequent model corrects the mistakes made by previous models iteratively. As a result of this iterative process, the model is continuously modified, and errors are minimised using gradient descent. First of all, this model creates a single leaf node rather than a tree, serving as an initial prediction for all samples. For continuous regression tasks, this initial estimate typically represents the average value of the target variable. GBR then generates future decision trees based on the prior tree's errors (residuals), continually enhancing forecasts. These decision trees typically have a small number of leaves, typically ranging from 4 to 20.

The algorithm continuously improves its predictions by iteratively constructing new regression trees that target the negative gradient of the loss function. This process effectively reduces the residual errors, or the discrepancies between the predicted values and the actual target values. The negative gradient is a powerful tool for optimisation, as it allows us to efficiently search for the parameters that minimise the loss function. GBR aims to minimise the loss function of the error metric.

Consider a data set

$$\mathrm{D}(\mathrm{x_i}, \mathrm{y_i})\mathrm{i} = 1 \text{ to } \mathrm{n} \tag{4}$$

Here,

x_i is the independent variables

y_i is the dependent variable (target value)

The first step involves building a single-leaf tree, based on the average of the target variable across all samples. Additionally, we utilize the loss function as an input to the algorithm. This loss function represents the discrepancy between the predicted values and the actual target values, ensuring that the predictions closely align with the actual values. Here, we take root mean squared error (RME) loss function for continuous regression tasks.

$$L = \frac{1}{n}\sum_{i=0}^{n}(y_i - \gamma_i)^2 \tag{5}$$

Here,

y_i is the dependent variable

γ_i is the predicted value

Now, we differentiate it with respect to γ and equate it to 0.

$$\frac{dL}{d\gamma} \sum {Y_i}^2 - 2Y_i\gamma_i + {\gamma_i}^2 = -2 \sum(Y_i - \gamma_i) \tag{6}$$

Calculating the pseudo residuals based on predictions from previous trees.

Mathematically as below:

$$r_{im} = -\left[\frac{\partial L\left(y_i, F(x_i)\right)}{\partial F(x_i)}\right] F(x) = F_{m-1}(x)\ for\ i = 1, \ldots, n. \tag{7}$$

Here,

m = m-th number of decision tree

F (x_i) = is the predictions made by previous tree, we equate m to m-1 to indicate the previous tree.

We are again taking a derivative of loss function and we get,

$$(y_i - \gamma_i) \tag{8}$$

Here, predicted value (γ_i) comes from previous tree.

Now, build a tree based on number of leaves restricted to predict the residuals by using the below equation.

$$\gamma_m = \arg\min \textstyle\sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma) \tag{9}$$

This is the loss function to minimize the residuals.

In the last step, to determine the actual predicted value in each leaf, the predicted value from the previous tree is combined with the pseudo residuals multiplied by the learning rate. We will run the loop for M trees basically for 100 tresses for better predictions.

In a nutshell, the mathematical formulation of the model is given below:

$$P_{GBR}^{t,g} = Model_i\,(X_{t-1}^{g}, X_{t-2}^{g}, \ldots\ldots, X_{t-W}^{g})$$
$$P_{GBR}^{t+1,g} = Model_{i+1}\,(X_{t}^{g}, X_{t-1}^{g}, \ldots\ldots, X_{t-W+1}^{g})$$
$$P_{GBR}^{t+w-1,g} = Model_{i+w-1}\,(X_{t+w-2}^{g}, X_{t+w-3}^{g}, \ldots\ldots, X_{t-1}^{g}) \quad (10)$$

Where,

W= window width (the number of steps ahead)

i=the current trained model

t=timestamp of the prediction

g= aggregation granularity of the data

It needs to be notified that all the equations from 4 to 10 mentioned above has been derived from (Houda et al.,2022).

3.3 Neural network architecture

A neural network is an artificial intelligence model that attempts to imitate the functioning of the human brain. It operates by establishing connections between processing elements, which serves as a metaphor for information processing in the brain. It is a collection of algorithms, generally patterned after the human brain, specifically developed to identify and comprehend patterns (Islam at el., 2019).

Klimasauskas (1989) mentioned that, at the fundamental level, a biological brain consists of a huge collection of neurons. Each individual neuron receives electrical and chemical impulses as inputs through its many dendrites and transmits the resulting signals through its axon. Axons establish connections with other neurons at specialised junctions known as synapses, where they transmit their output signals to other neurons, repeating this cycle numerous times.

An artificial neural network, adopting inspiration from the brain, is a network of interconnected components, known to as neurons. Neurons are capable of transmitting signals to one another through their connections. Every link is assigned a numerical value that indicates the weight or power of the transmission. Figure 3 is the illustrations of the human neuron structure. Artificial neural networks are computational models that mimic the operations of the human nervous system (Murata et al.,1992).

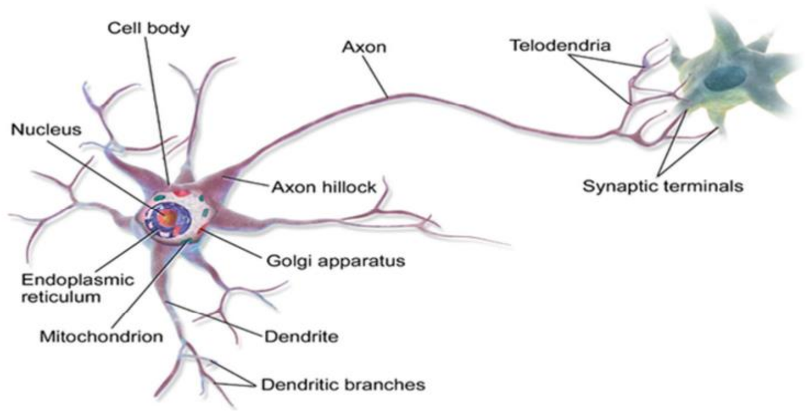

Figure 3. Structure of human neuron (Islam at el., 2019)

3.3.1 Recurrent Neural Network (RNN)

Recurrent neural networks (RNNs) are a type of artificial neural network that is specifically designed to handle sequential data. This makes them well-suited for tasks such as time series forecasting, language modelling, and machine translation. Therefore, RNN - based algorithms are highly effective at managing time-series data. They demonstrate exceptional proficiency in forecasting sequences by effectively retaining crucial information from both present and previous data points. This highly valuable for datasets that contain a time-series attribute (Paradari & Nordström.,2020).

3.3.1.1 Formulation of recurrent neural network (RNN)

RNNs are composed of a recurrent loop of interconnected neurons, enabling them to store and access information from previous inputs. This inherent memory has been indispensable for processing sequential data, where the current input is heavily influenced by past ones. The basic unit of an RNN is the recurrent unit. Each recurrent unit within an RNN receives both the current input and the hidden state from the previous time step.

It then processes this combined information to produce an updated, hidden state. This updated hidden state of an RNN is a vector of values that represents the network's current state. This model learns to predict the future values of a time series by entering its previous values into a neural network. The neural network modifies its internal connections based on the prediction of real future values. Before training the model, the user must define specific parameters, such as the number of neural network layers and nodes, as well as when training should be stopped. This figure illustrates the internal structure of the recurrent neutral network (Paradari & Nordström.,2020).

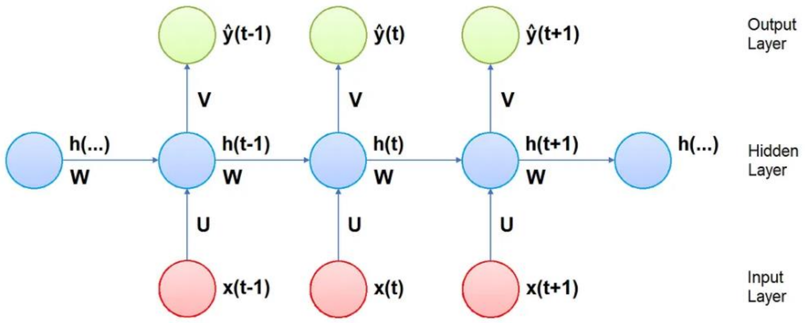

Figure 4. Internal structure of the RNN model (Pra,2020)

In the RNN model depicted in the figure 4, the input data at a given time step is connected to the hidden layer neurons of that time step using a weight matrix called U. The hidden layer neurons are also connected to the hidden layer neurons of the previous and next time steps using another weight matrix called W. Finally, the hidden layer neurons are connected to the output neurons of that time step using a weight matrix called V. All of these weight matrices remain constant throughout the entire training process. The vector x(t) is the input of the network at time step t. The vector h(t) is a hidden state at time t and is a sort of memory of the network. The calculation of the hidden state based on the current input and the previous time steps of the hidden state is given below:

$$h(t) = \tanh(Wh(t-1) + U_x(t)) \tag{11}$$

The vector $\hat{y}(t)$ is the output of the network at time t.

$$\hat{y}(t) = softmax(V_s(t)) \tag{12}$$

The main objective of this learning process is to find the best weight matrices U, V and W which makes the best prediction of y^(t), starting from the input x(t), of the real value y(t). To minimize the error between the real and the predicted values on the overall training set we calculate the loss function J.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{N_i} L(\hat{y}(t), y(t)) \tag{13}$$

Here,

The cost function L evaluates the distances between the real and predicted values on a single time step;

m is the size of the training set;

θ is the vector of model parameters.

To minimise the loss function, two major steps are used, such as forward propagation and backward propagation through time. These two steps are repeated many times, and the number of repetitions is known as the epoch number. It needs to be notified that all the equations from 11 to 13 mentioned above has been derived from (Pra,2020).

3.3.2 Gated recurrent units (GRU)

The gated recurrent units (GRUs) are a gating method in recurrent neural networks that was established in 2014 (Abumohsen at el., 2023). This model structure is simpler compared with another recurrent network. Furthermore, it is considered to be an enhanced version of LSTM with a simplified architecture, and has gained significant interest in numerous applications. The most significant applications are natural language processing (NLP), time series prediction such as stock price, weather and sales forecasting, speech recognition and synthesis (Wang et al., 2019).

3.3.2.1 Formulation of gated recurrent units (GRU)

The Gated Recurrent Unit (GRU) is a model based on Recurrent Neural Networks (RNN) that contains two internal gates. The gates are reset and update gates. The update gate determines whether the cell state should be modified using the current activation value. The reset gate is utilised to determine the significance of the previous cell state (Pedamallu, 2024).

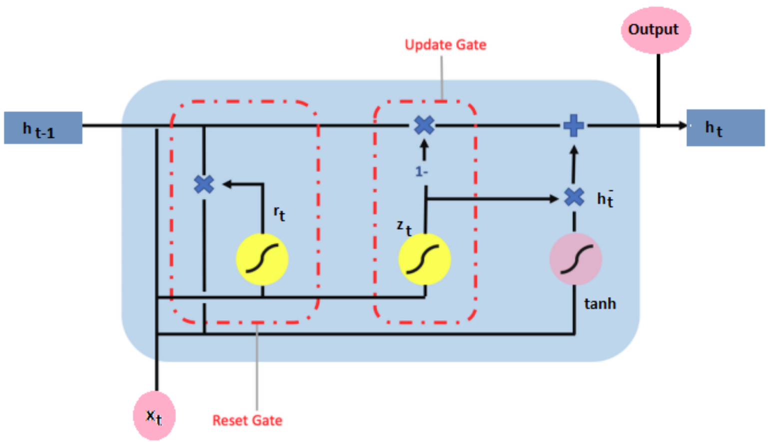

Figure 5. Basic structure of the GRU model (Pedamallu, 2024)

Figure 5. illustrates the fundamental architecture of the gated recurrent units. Here, the variable Xt represents the input at time t. Variables ht and ($\overline{h_t}$) are information vectors that indicate the temporary output and the hidden layer output at a specific moment, respectively. The sigmoid and tanh activation functions are denoted by (X) and tanh (x), respectively (Abumohsen at el., 2023). In this model, zt represents the update gate determining how much information should be brought to the next state cell. For the update gate, a bigger value means that more information is brought to the next state cell. rt represents the reset gate determining how much former information should be ignored. For the reset gate, a bigger value means that more information from the former cell may be ignored.

Following this process, the activation ht of the GRU at time step t is a linear interpolation between the previous activation ht-1 and the candidate activation ($\overline{h_t}$). The mathematic formulation of 14 to 17 are described GRU (Wang et al., 2019).

$$z_t = \sigma\,(W^z x_t + V^z h_{t-1} + b_z) \qquad (14)$$

$$r_t = \sigma\,(W^r x_t + V^r h_{t-1} + b_r) \qquad (15)$$

$$\widetilde{h_t} = tanh\,(W^c x_t + V^c(r_t \otimes h_{t-1})) \qquad (16)$$

$$\widetilde{h_t} = (1 - z_t) \otimes h_{t-1} + z_t \otimes \widetilde{h_t} \qquad (17)$$

In a GRU, when the reset value is near to 0, the model overlooks previous hidden state, as it is not important for future predictions. If the update gate is near to 1, then the information in that unit is copied over multiple steps. This setting primarily maintains control over how much historical data will be preserved (Pedamallu, 2024).

3.3.3 Long short-term memory (LSTM)

Long Short-Term Memory (LSTM) is a specialized recurrent neural network (RNN) specifically designed for processing sequential data, particularly time series information. With LSTM, long-term trends in data can be captured by using memory units with updates from the previous hidden state. Additionally, its feedback loops at each neuron, enabling it to process information more effectively and accurately. Moreover, the internal memory unit and gate mechanism reduced the vanishing gradient problems of the traditional RNN (Mahjoub et al.,2022).

3.3.3.1 Formulation of long short-term memory (LSTM)

LSTM architecture is fundamentally composed of memory cells, which are specifically designed to retain and access information over extended time intervals. Each memory cell consists of a cell state and three gates: an input gate, an output gate, and a forget gate. A LSTM unit is created by cell states, which convey information throughout the whole sequence and represent the network's memory. Each unit cell contains a forget gate, which determines what is relevant to keep from previous time steps; an input gate, which decides what information is relevant to add or how much new information is added to the cell state at the current time step; and an output gate, which determines the value of the output at the current time step (Pra, 2020).

These three gates are mainly control the maintenance and the update of information of the cell status. Following figure 6. shows the structure of an LSTM unit (Mahjoub et al.,2022).

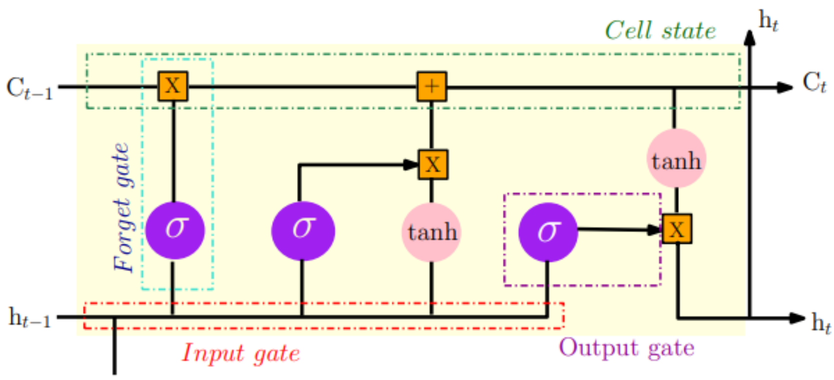

Figure 6. Internal structure of the LSTM model (Mahjoub et al.,2022)

The mathematical process is given below:

$$f_t = \sigma(w_f[h_{t-1}, X_t] + b_f) \tag{18}$$

$$i_t = \sigma(w_i[h_{t-1}, X_t] + b_i) \tag{19}$$

$$o_t = \sigma(w_o[h_{t-1}, X_t] + b_0) \tag{20}$$

$$a_t = tanh(w_a[h_{t-1}, X_t] + b_a) \tag{21}$$

$$c_t = f_t * c_{t-1} + i_t * a_t \tag{22}$$

$$h_t = o_t * \tanh(c_t) \tag{23}$$

Here, σ is the sigmoid activation function and it can be defined as:

$$\sigma(x) = (1 + e^{-x})^{-1} \tag{24}$$

Here,

f_t is the output values of the forget gate

f_t is the output values of the forget gate

i_t is the output values of the input gate

o_t is the output values of the output gate

c_t is the memory cell

X_t is the input vector at time t.

W_f, W_i, W_a, W_o are the weights matrices and b_f, b_i, b_a, b_o are the bias vectors. Mahjoub et al. (2022) introduced equation 18 to 24 as a mathematical representation of the LSTM model.

4 Statistical measurement for consumption

In statistic, the term correlation is used to explain the possible linear connection between two continuous variables (Mukaka, 2012). It is a measure of the monotonic connection between two variables. The main characteristic of monotonic relationship is when value of one variable increase same way other also increase or value of one variable increase, the other will be decreases (Schober et al., 2018). The relation between two variables are determined by the correlation coefficient. In general, there are two kinds of correlation coefficients: the spearman's rank correlation coefficient and the Pearson's product moment correlation coefficient. The uses of correlation coefficient depend on the types of variables are used in studies. Correlation coefficient is a dimensionless quantity also ranges from +1 to -1.

4.1 Correlation coefficient

In correlation coefficient, the zero value denotes that there is no relation between the two variables. Conversely, the positive correlation coefficient +1 indicates a positive relationship between the variables means when one variable increase other will be increase in the same way. Additionally, the negative correlation coefficient -1 indicates an inverse relationship between the variables means when one variable increase other will be decrease. In order to appropriately interpret the correlation between two variables, Schober et al., (2018) listed several ranges, including 0.0 to 0.20 for very weak correlation, 0.20 to 0.40 for weak correlation, 0.40 to 0.60 for moderate correlation, 0.60 to 0.80 for strong correlation, and 0.80 to 1.0 for very strong correlation.

4.1.1 Pearson correlation coefficient

The Pearson correlation coefficient is used to determine the strength and direction of the linear relationship between two variables when they are normally distributed with each other. On the other hand, the extreme values of the data also effect the strength and direction of the linear relationship between two variables. To calculate the Pearson correlation coefficient between two variables x and y by using given equation 25 (Campbell, 2021).

$$\mathrm{r} = \frac{\sum(\boldsymbol{x}-\overline{\boldsymbol{x}})(\boldsymbol{y}-\overline{\boldsymbol{y}})}{\sqrt{(\sum(\boldsymbol{x}-\overline{\boldsymbol{x}})^2 \sum(\boldsymbol{y}-\overline{\boldsymbol{y}})^2}} \tag{25}$$

Where,

r is the Pearson correlation coefficient

x and y are the two variables

$\bar{x}$ and $\bar{y}$ are the mean of x and y, respectively

4.1.2 Spearman's correlation coefficient

Spearman's correlation coefficient is used in the nonnormally distributed continuous data, ordinal data or data with relevant outliers (Schober et al., 2018). It is basically used for monotonic but nonlinear data. In the case of the spearman correlation calculation, it is basically calculated with the rank of the values of two variables instead of using the actual values of the variables (Kutner et al., 2005). Furthermore, this correlation coefficient is robust to non-normal data rather than being more vulnerable to outliers. To calculate the Spearman's correlation coefficient between two variables x and y, equation 26 is used (Schober et al., 2018).

$$r_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{26}$$

Where,

r_s is the Spearman's correlation co-efficient

x and y are the two variables

n is the number of data

di is the difference in ranks for x and y.

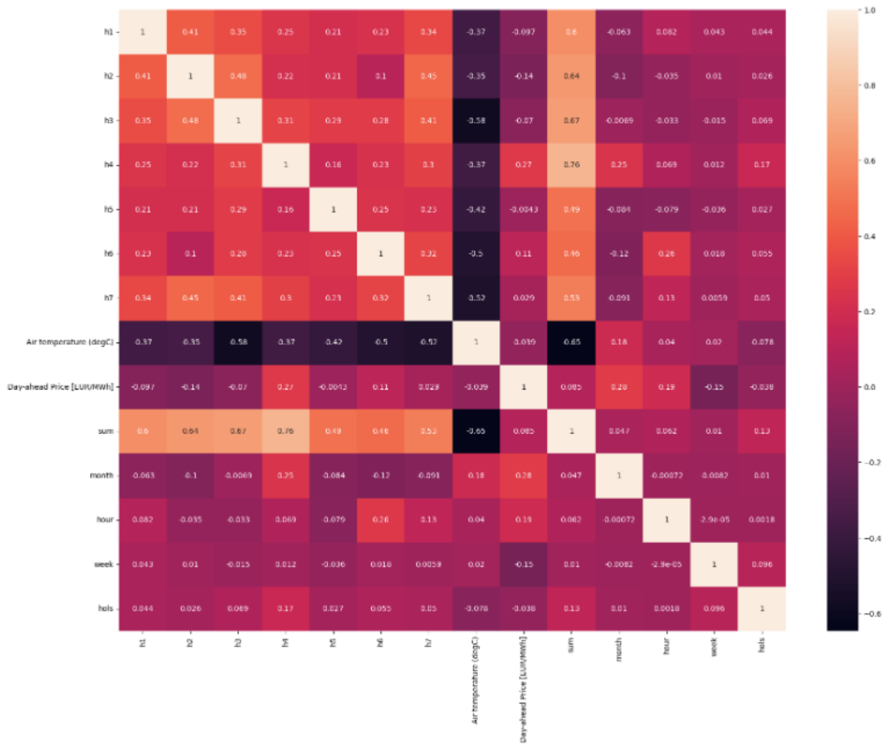

Figure 7. Heat map based on correlations coefficient between features

Figure 7 illustrates a heat map that visually represents the correlation between several features. The colour of each cell corresponds to the correlation coefficient, which might range from -1 to 1. The X and Y axes both indicate an identical set of features, which include: Hourly data points (h1, h2, ..., h7) include environmental and temporal data, including air temperature, day-ahead pricing, sum, month, hour, week, and holidays (hols). The bar displays a slope ranging from deep purple to a light orange.

Dark Purple (~-0.6 to -0.7): Indicates strong negative correlations. Red (~0): Indicates an association that is neutral or insignificant. Light orange colour, with a value range of approximately 0.8 to 1, indicates the presence of strong positive correlations.

4.2 Correlations between consumption, price, and temperature

For better electricity consumption, it is crucial to know the correlations between consumption, price, and temperature, as they also have an impact on energy companies, policymakers, and consumers. From the perspective of consumers, there is a strong correlation between consumption and price, and they can easily shift their consumption time from peak demand periods to off-peak periods. Moreover, the price of electricity depending on the day typically higher during peak our and lower in off peak hour. Since renewable energy is intermittent in nature, at times of higher demand, electricity is produced from fossil fuels. In addition, demand response programs increase grid stability and inform customers of the costs that come with using electricity during peak and off-peak hours.

4.2.1 Relationship between consumption and day-ahead price (EUR/MWH)

House 1: The Pearson correlation coefficient is -0.097 indicates a negative relationship between consumption and price. This suggests that there is a small inverse relationship between the two variables, i.e., consumption decreases as prices rise. On the other hand, the Spearman correlation coefficient is -0.13, which also shows a negative relationship between consumption and price, which means that when price increases, consumption decreases slightly.

House 2: The Pearson correlation coefficient is -0.14, also mentioned a weak negative relationship between consumption and price. This indicate that there is a slight inverse relationship between the two variables, meaning that as price increases, consumption decreases. Conversely, the value of Spearman correlation coefficient is -0.17 indicates a negative connection between price and consumption, meaning that a rise in price is associated with a slight drop in consumption.

House 3: The Pearson correlation coefficient is -0.07 also indicates a weak negative relationship between consumption and price. This means that there is a slight inverse relationship between the two variables, meaning that as price increases, consumption decreases. Conversely, the value of Spearman correlation coefficient is -0.13 indicates a negative connection between price and consumption, meaning that a rise in price is associated with a slight drop in consumption.

House 4: The Pearson correlation coefficient of 0.27 indicates a slightly strong positive relationship between consumption and price. This means that there is a positive relationship between the two variables, meaning that as price increases, consumption also increases. On the contrary, the Spearman correlation coefficient of 0.0079 also shows a weak positive relationship between consumption and price. It also mentions a positive connection: as price increases, consumption also increases slightly.

House 5: The Pearson correlation coefficient is -0.0043 that indicates a weak relationship between consumption and price. This means that this relationship between the two variables not bring any significant change. The Spearman correlation coefficient is -0.031 also mentioned a weak negative relationship between consumption and price also mentioned as price increases, consumption decreases slightly but not in strong way.

House 6: The Pearson correlation coefficient of 0.11 also indicates a weakly beneficial relationship between consumption and price. This means that there is a slightly positive relationship between the two variables, meaning that as price increases, consumption increases. The Spearman correlation coefficient of 0.085 indicates a weak positive relationship between consumption and price. It is also mentioned that as price increases, consumption increases slightly.

House 7: The Pearson correlation coefficient is 0.029, which indicates a weakly positive relationship between consumption and price. This means that there is a slightly moderate relationship between the two variables, meaning that as price increases, consumption also increases, but not significantly. The Spearman correlation coefficient of 0.0098 shows a weak positive association between consumption and price. It is also mentioned that as price increases, consumption increases slightly.

Based on the value of Pearson and Spearman correlation coefficient, the house 2 is more flexible then other.

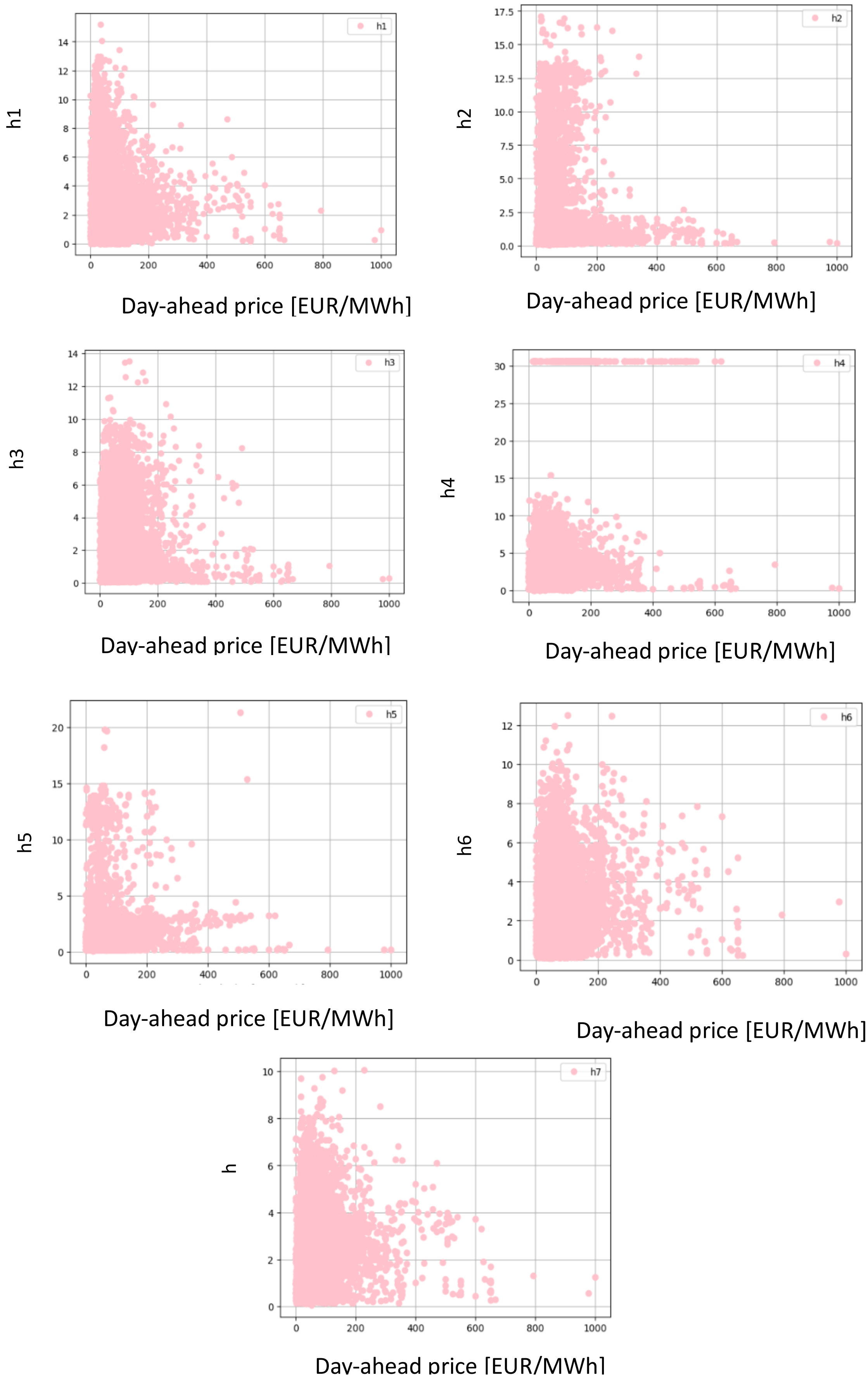

Figure 8. Consumption based on day-ahead price [EUR/MWh]

4.2.2 Relationship between consumption and air temperature (deg C)

House 1: The Pearson correlation coefficient of -0.37 indicates a moderate negative association between consumption and air temperature. This means that there is a slight inverse relationship between the two variables, meaning that as temperature increases, consumption decreases, or vice versa. The Spearman correlation coefficient of -0.51 also indicates a strong negative relationship between consumption and air temperature. As temperature increase, consumption also decreases.

House 2: The Pearson correlation coefficient of -0.35 mentioned a weak negative relationship between consumption and air temperature. As a result, there is a slightly inverse association between the two variables, meaning that as air temperature increases, consumption decreases. The Spearman correlation coefficient of -0.13 mentioned a weak negative relationship between consumption and air temperature; it also mentioned that as air temperature increases, consumption decreases insignificantly.

House 3: The Pearson correlation coefficient of -0.58 mentioned a strong negative relationship between consumption and air temperature. Consequentially, there appears to be a strong inverse relationship between the two variables, meaning that as air temperature increases, consumption decreases, or vice versa. The Spearman correlation coefficient of -0.61 maintains a strongly negative relationship between consumption and air temperature, and as air temperature increases, consumption decreases significantly.

House 4: The Pearson correlation coefficient of -0.37 mentioned a strong negative relationship between consumption and air temperature. Thus, it would seem that the two variables have a strong inverse this regard, i.e., that consumption reduces as air temperature grow or vice versa. The Spearman correlation coefficient of -0.61 maintains a strongly negative relationship between consumption and air temperature, and as air temperature increases or, consumption decreases significantly.

House 5: The Pearson correlation coefficient of -0.42 mentioned a strong negative relationship between consumption and air temperature. Consequently, a significant negative connection is present between the two variables, indicating that consumption falls or rises when the air temperature also rises or falls. The Spearman correlation coefficient of -0.83 indicates a very strong negative relationship between consumption and air temperature. It is also mentioned that consumption depends on the air temperature and data are more predictable in this situation. For heating or cooling the house, consumers use more electricity. As the air temperature increases or decreases in that situation, consumers are likely to adjust their usage habits immediately.

House 6: The Pearson correlation coefficient of -0.50 also maintains a strong negative relationship between consumption and air temperature. This means that there is a significant inverse relationship between the two variables, meaning that as temperature increases, consumption decreases, or vice versa. The Spearman correlation coefficient of -0.60 indicates a strong negative relationship between consumption and air temperature. This value indicates that in the colder months, people use more electricity for heating purposes, and in the warmer months, people use less electricity.

House 7: The Pearson correlation coefficient of -0.52 also mentioned a strong negative relationship between consumption and air temperature. This shows a significant inverse relationship between the two variables, meaning that as temperature increases, consumption decreases, or vice versa. The Spearman correlation coefficient of -0.60 shows the interconnection between consumption and temperature, as these two variables are more dependent on each other means that when temperature increase then consumption decreases or vice versa. In the cooler months, people use more electricity to warm up their houses, and in the warmer months, they use less electricity.

Based on the value of Pearson and Spearman correlation coefficient, the house 3 and 5 are more flexible then other.

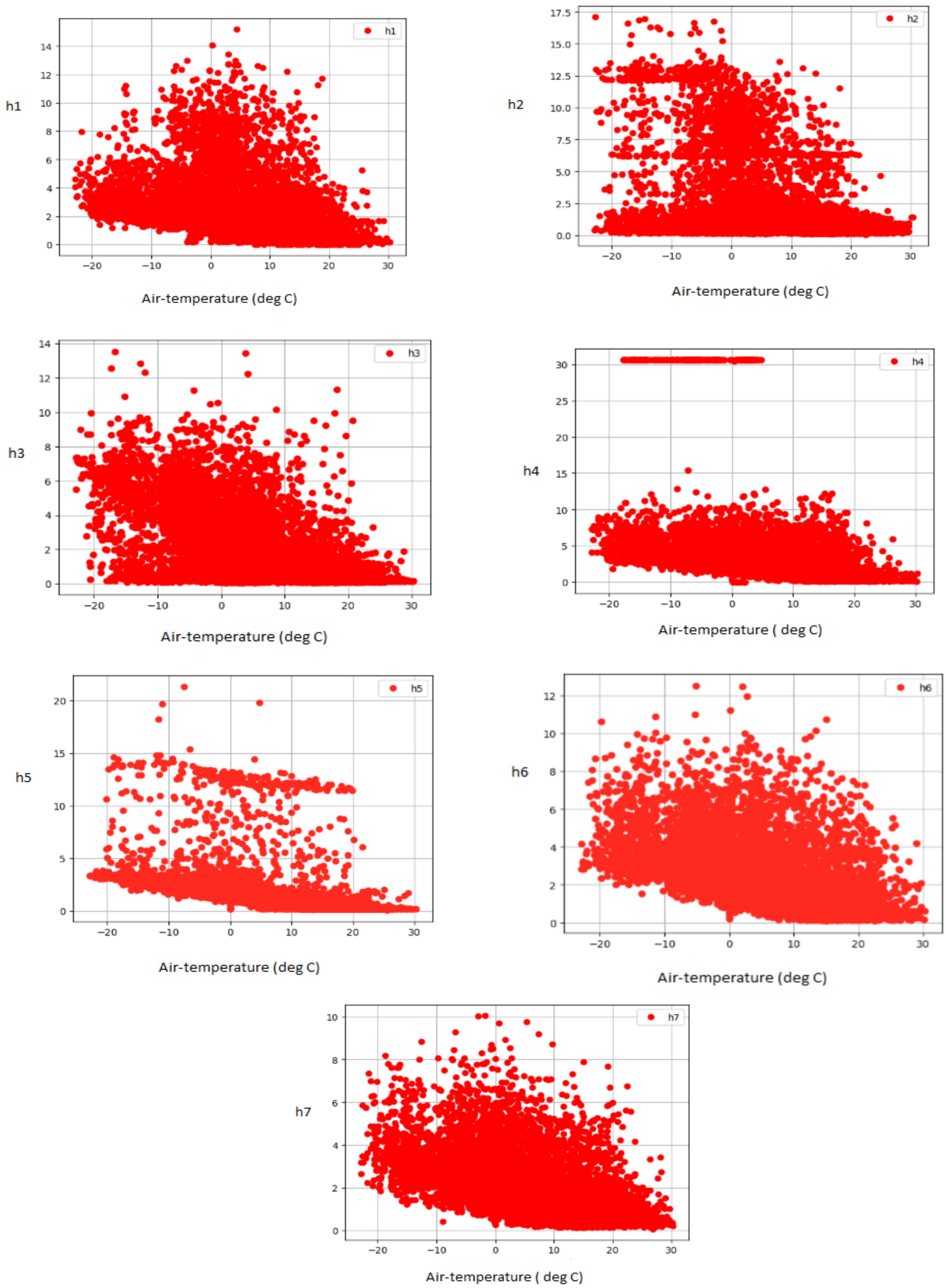

Figure 9. Consumption based on air-temperature (degree Celsius)

4.3 Insights into house flexibility

According on the correlations, the houses have different degrees of flexibility on their consumption patterns based on price and temperature changes.

House 1: The consumer of house 1 is also flexible, with the strongest negative correlation between consumption and temperature, but on the other hand, less flexible on price. The value of these correlation co-effects illustrates that the consumer of the house is able to adjust its consumption patterns to a greater extent in response to changes in temperature, not in price.

House 2: The consumer of house 2 is slightly more flexible than house 1, because the consumer of this house is more sensitive to temperature and price. Furthermore, the consumer of house 2 is able to adjust its consumption patterns to a greater extent in response to changes in price. When the price increases, they are able to reduce their consumption or organize their work based on the price.

House 3: The consumer of house 3 is the least flexible compared with house 2 on the basis of price but more flexible with air temperature compared with all other houses, with the strongest negative correlation between consumption and air temperature. Consumers in this house are more sensitive to temperature. Moreover, when temperatures go down or rise, they increase or decrease the use of electricity.

House 4: The consumer of house 4 has a positive correlation coefficient with price, indicating that price changes have no influence on their consumption, indicating that they are not price flexible. Furthermore, the consumption pattern of this house is more flexible with the air temperature; increasing and decreasing the temperature creates a moderate effect. In the same way, they reduce consumption as the temperature rises and increase it when temperatures decrease.

House 5: The consumption of house 5 has no significant correlation with price but also a moderately negative correlation with air temperature. The house's consumption has no influence of price, indicating that it is not price-flexible. In addition, this house is less flexible than others.

House 6: The consumer of house 6 is less flexible with price but more with air temperature.

House 7: The consumer of house 6 is less flexible with price but more with air temperature.

4.4 Reasons behind varying degrees of flexibility

According to Reiss & White (2005) mentioned, there are number of factors are responsible for the varying degrees of flexibility throughout house such as:

Household composition: The number of people in the house creates a significant impact in the case of consumption; more people with less flexibility mean it's difficult to adjust consumption habits based on changing circumstances.

Household income: People with higher income with more flexible patterns in consumption to adjusting their financial resources. In other words, there is a positive correlation between income and consumption, indicating that as income levels rise, so does the level of consumption. On the other hand, people with low income have negative correlation between income and consumption when price will be high they reduce their consumption.

Household temperature: Temperature also creates a significant impact in consumption. When temperature increase or decrease people uses more electricity. For example, during hot summer months, households may consume more energy for air conditioning purposes, while in colder winter months, they may consume more energy for heating their house.

Efficient appliance: In the case of consumption, houses with more efficient appliance consume less amount energy. More flexibility and an excellent operating principle allow efficient appliances to use less power.

5 Flexibility forecast and different time horizons

The importance of precise forecasting in today's constantly evolving environment cannot be exaggerated. Forecasts are of utmost importance in diverse industries and sectors, encompassing agriculture, transportation, energy production, and catastrophe preparedness. These forecasts enable companies and organisations to strategically plan their operations, optimise their activities, and mitigate potential hazards. For instance, within the realm of agriculture, precise weather predictions assist farmers in planning their planting and harvesting timetables, while also enabling them to predict potential infestations or illnesses that could impact their crops. Furthermore, precise projections are crucial for energy production, as they enable utilities to predict the demand for heating and cooling, optimise energy distribution, and eliminate power outages. This allows them to modify their production schedules and optimise the utilisation of resources such as sustainable energy sources like solar and wind power (Ssekulima et al., 2016).

Researcher Ssekulima et al. (2016) pointed out that accurate weather forecasts enable electricity producers to predict periods of high or low demand, optimise the utilisation of various power sources, and efficiently manage their energy resources to meet consumer needs. In addition, they could help in the prevention of blackouts or shortages by enabling power providers to prepare and make necessary arrangements for probable extreme weather phenomena, such as intense thunderstorms or heatwaves. Furthermore, precise predictions offer vital insights to diverse sectors and enterprises, enabling them to make well-informed choices and mitigate potential hazards. Accurate weather forecasts are crucial for companies and sectors to strategically plan their operations, optimise their activities, assure safety, and minimise hazards.

The recent advancement in the energy industry has resulted in the widespread utilisation of renewable energy sources (RESs) as a means to mitigate climate change. Renewable energy sources (RESs) exhibit high volatility as they rely heavily on weather conditions.

According to Khajeh et al. (2021), the power generated by renewable sources is unpredictable and highly variable. Consequently, the responsibility of maintaining a balance between power generation and load in the power system becomes more difficult, and the presence of renewable energy sources (RESs) raises the likelihood of security and stability problems. Presently, there is a substantial disparity between the highest and lowest levels of demand on the power grid, which is seen not just in established nations but also in emerging nations. This disparity leads to increased network losses and a reduced lifespan of equipment. From the perspective of the power system, there are multiple compelling reasons to urgently enhance the electrical demand flexibility of buildings. Initially, there is a growing percentage of renewable electricity being integrated into the electricity grid through the utilisation of wind and solar power, both of which are characterised by their intermittent nature. Furthermore, the occurrence of extreme weather conditions, such as intense heat or extreme cold, along with the phenomenon of climate change, exerts a substantial impact on the dependability and functioning of electrical components. Furthermore, the quantity and size of conventional fossil fuel power plants have been diminishing in developed nations due to the adoption of renewable energy sources. However, there is a simultaneous rise in electricity demand in emerging industrialised countries. Due to these factors, the issues around electric power balancing is getting increasingly difficult. Various strategies can be employed to enhance the electrical adaptability of grid-interactive buildings. Demand response (DR) is widely recognised as a primary method for achieving electrical flexibility and generating economic advantages through the utilisation of dynamic energy pricing and DR incentives. Additionally, DR has the potential to significantly enhance energy flexibility capacity (Chen et al.,2018).

The incorporation of intermittent renewable energy resources into power systems on a large scale necessitates a greater demand for flexibility services. In order to meet the flexibility requirements, system operators in the future must make use of a wider range of adaptable energy resources from all levels of the system. Customers who are aggregated as a local energy community (LEC) have the capacity to contribute a portion of the necessary flexibility. Precise prediction of the adaptable capabilities of an LEC is crucial in this context (Firoozi et al., 2021).

The forecasting needs for smart grids encompass various aspects, including wind and solar power generation, electricity consumption, energy pricing, energy flexibility, power system inertia, and electric vehicle charging demand. In the case of forecasting, time is a crucial variable that encompasses the specific hour of the day, the particular day of the week, and the specific month. Moreover, different types of forecasting like short-term projections are utilised in the scheduling of smart grids to achieve a balance between power generation and demand. This involves activating flexibility, dispatching generation units, and imposing limitations on renewable energy sources. On the other hand, long-term forecasts for developing smart grids encompass projections for infrastructure investments, strategies for integrating renewable energy sources, and the advancement of the market, among other considerations.

To address the problems outlined above, system operators must enhance the system's adaptability by including adaptable resources. Flexibility in power systems refers to the capacity to continually alter the operating point of the system and withstand anticipated and unanticipated changes in operating conditions. Therefore, the adaptable power system should promptly and cohesively adjust to fluctuations in both electricity use and production. Flexibility, in addition, refers to the capacity of generation- or demand-based flexible resources to adjust their behaviour during operation in response to external signals transmitted by grid operators (Khajeh et al.,2021).

This study primarily examines the residential demand response (DR) for aggregated customers, with a specific focus on market prices. Additionally, the impact of new technologies, such as various machine learning models, on improving the efficiency and coordination of demand response was also evaluated.

5.1 Data manipulation

The original pricing data from the test dataset (X_test) has been enhanced by adding 10 units and 20 units to each value, resulting in a new dataset (X_test_2). The model's predictions using the original dataset are labelled as (y_pred_test), whereas the predictions using the changed dataset are labelled as (y_pred_test_2).

5.2 Analysis of model flexibility

Flexibility in a model refers to the extent to which the model's output (prediction) varies when there are changes in the input. In order to determine the flexibility, we employed three models: recurrent neural network (RNN), gated recurrent unit (GRU), and Long Short-Term Memory (LSTM).

5.2.1 Flexibility calculation for recurrent neural network (RNN)

Analysing the adaptability of an RNN model by modifying input prices and assessing the impact on the model's forecasts. For the RNN model, the input is the price, and the output can be any financial measure that the model is forecasting, such as demand, cost, or another variable that depends on price.
Flexibility as the difference in predictions from the original input data and the modified input data, calculated by using given equation 28 (Pfeifer at el.,2021).

$$\text{Flexibility} = \text{y pred\_original} - \text{y pred\_modified} \tag{27}$$

$$\text{Flexibility} = \text{y\_pred\_test} - \text{y\_pred\_test\_2} \tag{28}$$

Flexibility test with increased prices by 10 units:
Applying the dataset, implement the RNN model. This graphic demonstrates the adaptability of a neural network model in its response to variations in input data. This axis displays the dataset index, which may consist of discrete data points arranged in a series

of time steps. Each index relates to a distinct occurrence where the model input had been modified and a prediction was generated.

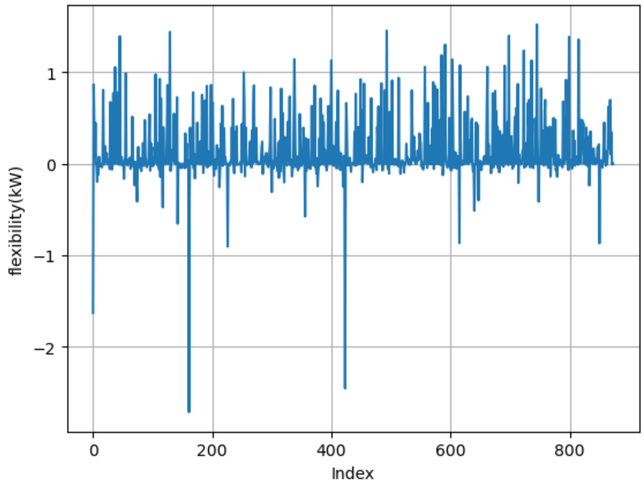

Figure 10. Flexibility curve based on data

The flexibility is measured in kilowatts (kW), indicating the scale of change in the model's output due to changes in the input.

Flexibility test for different house:

In order to determine the flexibility of various houses, it is necessary to examine the data of two specific houses, namely house 2 (h2) and house 4 (h4). Figure 11 depicts the flexibility values for house 2. The X-axis, denoted as h2, shows the index of data points in the dataset. Every point on the axis represents a particular event in the dataset when the model was given an input (most likely the price) and generated a forecast. The Y-axis depicts the flexibility, measured in kilowatts (kW), which measures the model's output change in reaction to an input variation, namely an increase in price.

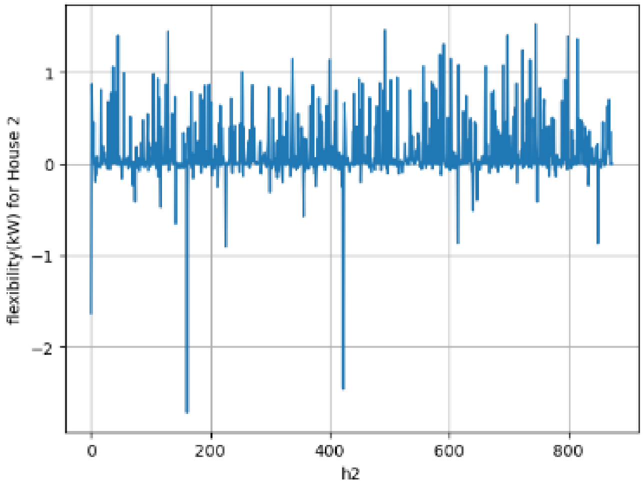

Figure 11. Flexibility for house 2 (h2)

Figure 12 indicates the flexibility values associated with house 4. The X-axis, denoted as h4, shows the index of data points in the dataset. Every point on the axis represents a distinct occurrence in the dataset when the model was given an input (most likely the price) and generated a forecast. The Y-axis depicts the flexibility, measured in kilowatts (kW), which measures the model's output change in reaction to an input variation, namely an increase in price.

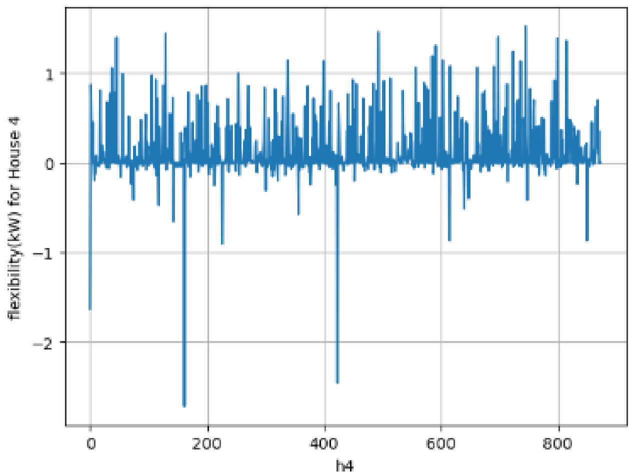

Figure 12. Flexibility for house 4 (h4)

Flexibility test with increased prices by 20 units:

Execute the model using the data set as input. This graphic demonstrates the adaptability of a neural network model in its response to variations in input data. The axis indicates the index of the dataset, which can refer to either time steps or individual data points within a series. Every index relates to a distinct occurrence in which the model input was changed and a prediction was generated.

The flexibility is measured in kilowatts (kW), indicating the scale of change in the model's output due to changes in the input.

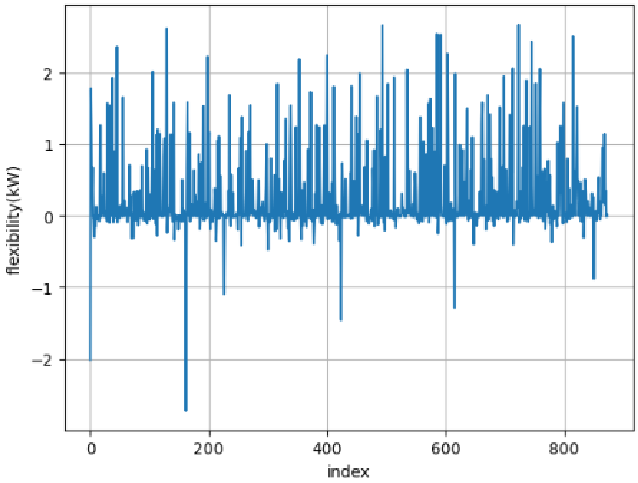

Figure 13. Flexibility curve based on data

Flexibility test for different house:

In order to assess the flexibility of various houses, it is necessary to examine the data of two specific houses, namely house 2 (h2) and house 4 (h4). Figure 14 depicts the flexibility values for house 2 when the price is increased by 20 units compared to the prior price. The X-axis, denoted as h2, shows the index of data points in the dataset. Every point on the axis represents a particular event in the dataset when the model was given an input (most likely the price) and generated a forecast. The Y-axis depicts the flexibility, measured in kilowatts (kW), which measures the model's output change in reaction to an input variation (probably an increase in price).

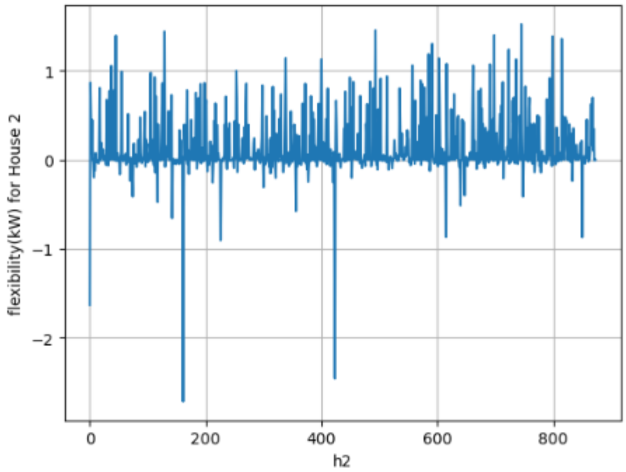

Figure 14. Flexibility for house 2 (h2)

This figure.15 illustrates that the flexibility values for house 4 when the price is 20 unit more than the previous one.

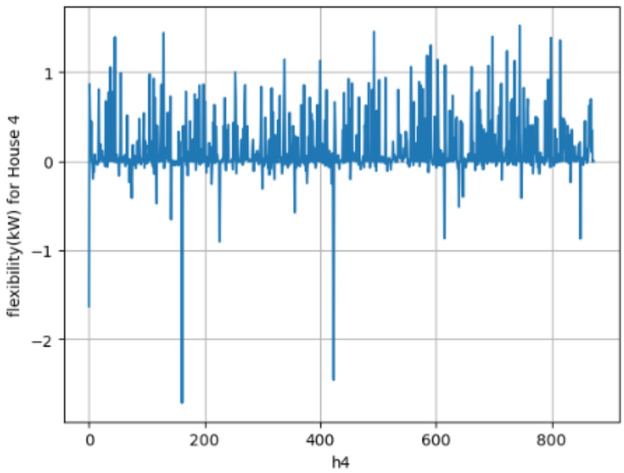

Figure 15. Flexibility for house 4 (h4)

Sensitivity to price changes:

In this instance, the flexibility values change both above and below zero. The positive values on the flexibility plot indicate that there is a positive difference between the initial prediction and the changed prediction. In this scenario, it is evident that increasing the input variable (price) leads to a decline in the model's forecast (demand).

A positive flexibility number signifies an inverse relationship between price and demand, where an increase in price leads to a drop-in demand. This feature demonstrates a common economic behaviour known as the inverse relationship between price and consumption, where people decrease their consumption as prices rise. In a comparable manner negative values represent that the initial forecast is lower than the prediction generated based on the updated input. This is a consequence of the model generating greater values as the input is increased. This demonstrates a negative correlation, where the demand (output) increases as the price (input) increases. This refers to a common economic phenomenon in which people tend to increase their consumption as prices rise. In traditional market dynamics, this may seem contradictory unless the desired outcome is something that benefits from higher prices, such as revenue or an increase in the use of alternative resources when the primary resource gets too expensive. Furthermore, the distribution of values around zero in both plots does not exhibit any apparent inclination towards positive or negative values. This suggests that the model does not consistently overestimate or underestimate with any significant sensitivity when price inputs are changed.

According to the flexibility plots, the model consistently responds to changes in input across multiple houses. However, the specific changes in output can vary, possibly due to variances in the underlying data characteristics or unique aspects of each house. Furthermore, it has been demonstrated that as prices rise, flexibility becomes more favourable. Conversely, when we raise the price by 10 units, both House 2 and House 4 exhibit comparable levels of adaptability, both in terms of their range and variability. The fluctuations are densely clustered around the zero line, exhibiting a comparable distribution of peaks and troughs in both instances. Based on the analysis, neither house has a distinct advantage in flexibility compared to the other, as shown by the presented graphs. Both houses exhibit identical levels of flexibility, indicating that their model outputs are equally responsive to changes in input.

Similarly, when the price is increased by 20 units, both houses likewise exhibit greater variability in response to the identical changes in input. House 2 has significant variations, although these changes are concentrated closer to the zero line and have fewer and less pronounced peaks.

House 4, on the other hand, demonstrates higher and more frequent peaks, indicating that the model's output is significantly influenced by changes in input.
Furthermore, House 4 exhibits a higher level of sensitivity in comparison to House 2. The higher amplitude and frequency of peaks in the flexibility plot for House 4 indicate that its model output is more responsive to variations in input conditions. This heightened sensitivity could be beneficial for forecasting.

5.2.2 Flexibility calculation for gated recurrent unit (GRU)

Evaluating the adaptability of a gated recurrent unit (GRU) model by manipulating the input (prices) and evaluating its impact on the model's forecasts. For the GRU model, the input is the price, and the output can be any financial measure that the model is forecasting, such as demand, cost, or another variable that depends on price. Utilise equation (28) to determine the flexibility of electrical consumption in various households.

Flexibility test with increased prices by 10 units:

Based on the dataset, the GRU model was applied. This graphic demonstrates the adaptability of a gated recurrent unit in its response to varying input data. The axis in concern indicates the index of the dataset, which can refer to either time steps or individual data points within a series. Every index relates to a distinct occurrence in which the model input changed and a forecast subsequently generated.

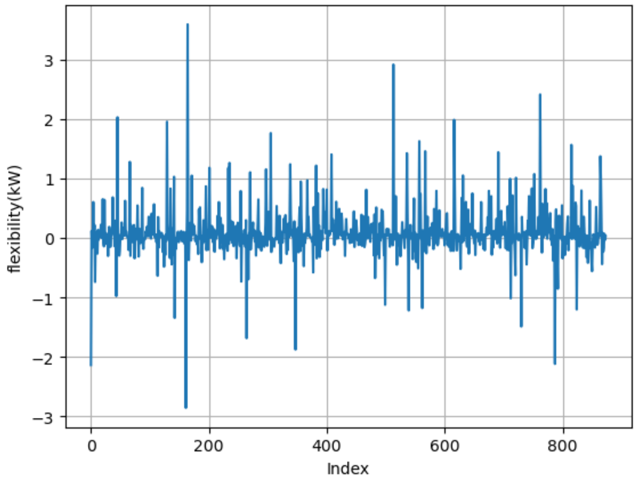

Figure 16. Flexibility curve based on data

The flexibility is measured in kilowatts (kW), indicating the scale of change in the model's output due to changes in the input.

Flexibility test for different house:

In order to assess the adaptability of various houses, examine the data of two specific houses, namely house 2 (h2) and house 4 (h4). This picture depicts the flexibility values for house 2. The X-axis, denoted as h2, shows the index of data points in the dataset. Each point on the axis represents a distinct event in the dataset when the model was given an input (probably the price) and generated a forecast. The Y-axis depicts the flexibility, measured in kilowatts (kW), which measures the model's output change in reaction to an input variation (often an increase in price).

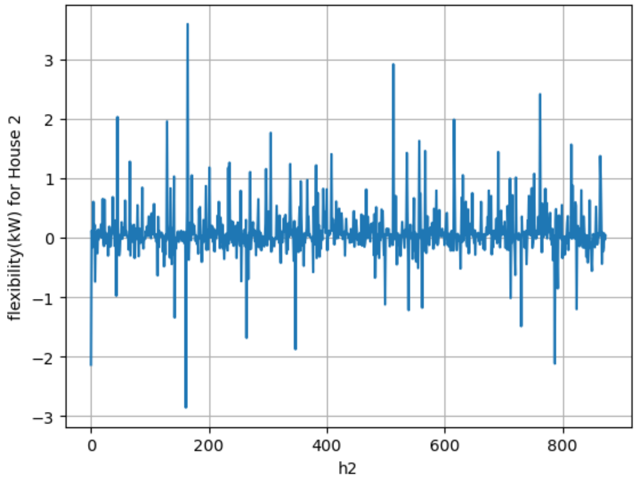

Figure 17. Flexibility for house 2 (h2)

Figure 18 depicts the flexibility values for house 4. The X-axis, denoted as h4, shows the index of data points in the dataset. Every point on the axis represents a distinct event in the dataset when the model was given an input (most likely the price) and generated a forecast. The Y-axis depicts the flexibility, measured in kilowatts (kW), which measures the model's output change in reaction to an input variation (often an increase in price).

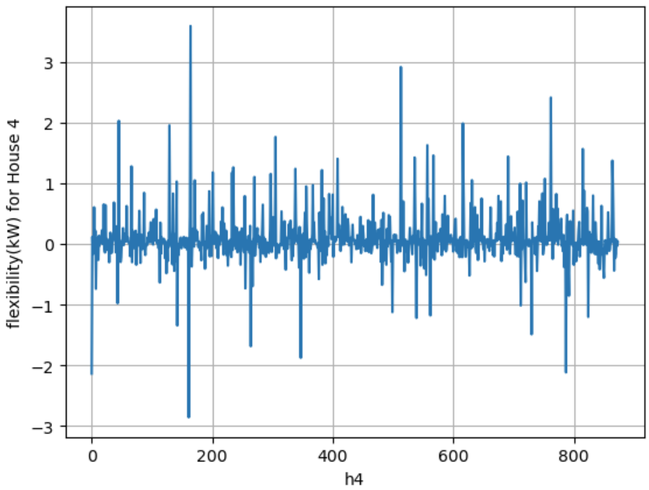

Figure 18. Flexibility for house 4 (h4)

Flexibility test with increased prices by 20 units:

Execute the model using the data set as input. This graphic demonstrates the adaptability of a gated neural network model in its response to variations in input data. The axis indicates the index of the dataset, which can refer to either time steps or individual data points within a series. Every index relates to a distinct occurrence in which the model input was altered and a prediction was generated.

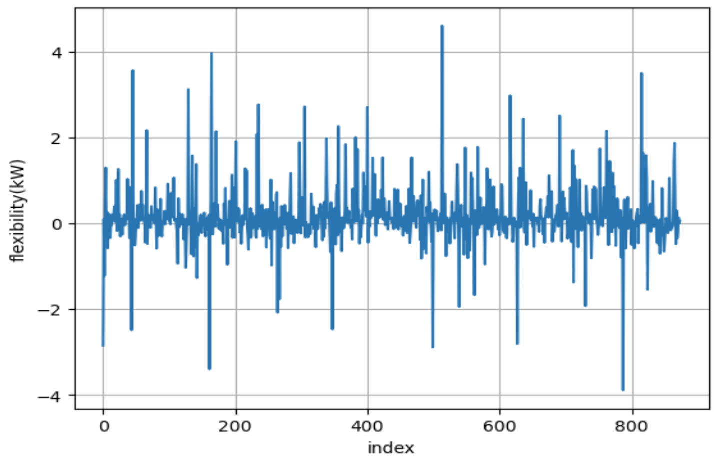

Figure 19. Flexibility curve based on data

The flexibility is measured in kilowatts (kW), indicating the scale of change in the model's output due to changes in the input.

Flexibility test for different house:

In order to assess the flexibility of various houses, it is necessary to examine the data of two specific houses, namely house 2 (h2) and house 4 (h4). Figure 20 depicts the flexibility values for house 2 when the price is increased by 20 units compared to the prior price. The X-axis, denoted as h2, shows the index of data points in the dataset. Every point on the axis represents a particular event in the dataset when the model was given an input (most likely the price) and generated a forecast. The Y-axis depicts the flexibility, measured in kilowatts (kW), which measures the model's output change in reaction to an input variation (often an increase in price).

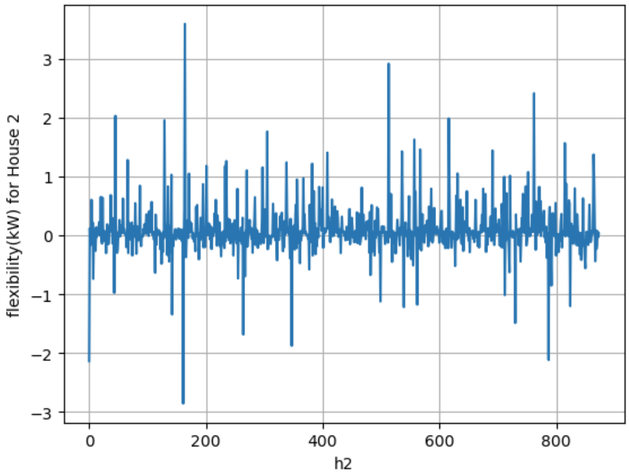

Figure 20. Flexibility for house 2 (h2)

This figure.21 illustrates that the flexibility values for house 4 when the price is 20 unit more than the previous one.

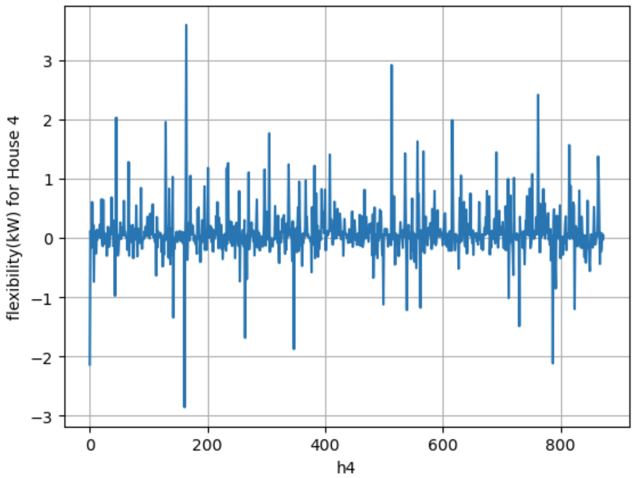

Figure 21. Flexibility for house 4 (h4)

Sensitivity to price changes:

In this instance, the flexibility values fluctuate both above and below zero. The positive values on the flexibility plot indicate that there is a positive difference between the initial prediction and the changed prediction. In this scenario, it is evident that augmenting the input variable (price) leads to a decline in the forecasted outcome (demand) of the model. A positive flexibility number signifies an inverse relationship between price and demand, where an increase in price leads to a drop-in demand. This characteristic demonstrates a common economic phenomenon known as the inverse relationship between price and consumer demand, where customers decrease their consumption as prices rise.

In a comparable manner, negative numbers signify that the updated input's prediction exceeds the initial estimate. This occurs as a consequence of the model making higher predictions as the input is increased. This demonstrates a negative correlation, where the demand for a product increases as the price of the product increases. This refers to a common economic conduct in which customers increase their consumption as prices rise. In traditional market dynamics, this may seem counterintuitive unless the desired outcome is something that benefits from higher prices, such as revenue or an increase in the use of alternative resources if the primary resource becomes too expensive.

Furthermore, the average distribution of values around zero in both plots does not exhibit any discernible inclination towards positive or negative values. This suggests that the model does not consistently overestimate or underestimate with any substantial partiality when price inputs are modified.

According to the flexibility plots, the model consistently responds to changes in input across multiple houses. However, the specific changes in output can vary, possibly due to variances in the underlying data characteristics or unique aspects of each house. Furthermore, it has been demonstrated that as prices rise, flexibility becomes more favourable. House 2 exhibits a significantly higher degree of flexibility compared to House 4.

The plot of House 2 has a higher frequency of approaching both the upper and lower limits of the flexibility range. This implies that the model output of House 2 is more responsive to changes in input prices, exhibiting larger peaks and deeper troughs, which indicates a higher level of flexibility. The plot exhibits substantial variability with multiple spikes, both positive and negative, indicating that the model's output for House 2 is very responsive to changes in input values. A greater prevalence of more prominent extreme values, both positive and negative, suggests a higher level of reactivity or adaptability to changes in input.

5.2.3 Flexibility calculation for long short-term memory (LSTM)

Evaluating the flexibility of a Long Short-Term Memory (LSTM) model through changing the input (prices) and evaluating the impact on the model's forecasts. For the LSTM model, the input is the price, and the output can be any financial measure that the model is forecasting, such as demand, cost, or another variable that depends on price. Apply equation (28) to determine the use of electricity in various households.

Flexibility test with increased prices by 10 units:

Apply the provided data set to run the LSTM model. This graphic demonstrates the adaptability of a Long Short-Term Memory (LSTM) model in its response to variations in input data. The axis indicates the index of the dataset, which may correspond to either time steps or individual data points within a series. Each index relates to a distinct occurrence where the model input changed and a prediction was generated.

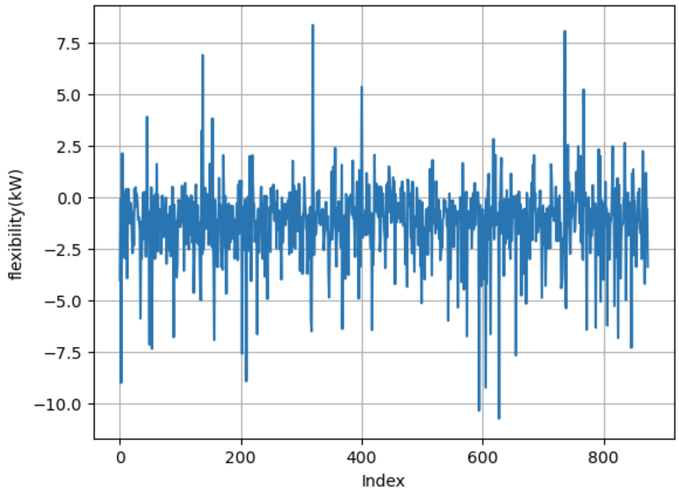

Figure 22. Flexibility curve based on data

The flexibility is measured in kilowatts (kW), indicating the scale of change in the model's output due to changes in the input.

Flexibility test for different house:

In order to evaluate the flexibility of different houses, examine the data of two specific houses, namely house 2 (h2) and house 4 (h4). The flexibility values for house 2 are shown in this figure, 23. The X-axis, denoted as h2, shows the index of data points in the dataset. Every point on the axis represents a particular event in the dataset when the model was given an input (probably the price) and generated a forecast. The Y-axis depicts the flexibility, measured in kilowatts (kW), which measures the model's output change in reaction to an input variation, namely an increase in price.

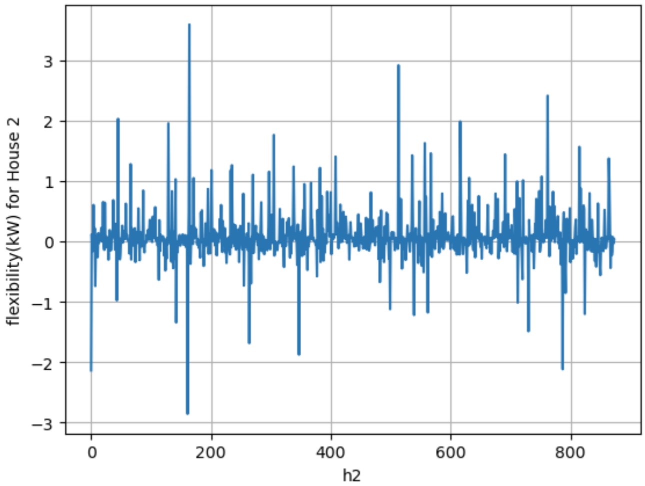

Figure 23. Flexibility for house 2 (h2)

Figure 24 shows the flexibility values for house 4. The X-axis, denoted as h4, shows the index of data points in the dataset. Every point on the axis represents a distinct occurrence in the dataset when the model was given an input (probably the price) and generated a forecast. The Y-axis depicts the flexibility, measured in kilowatts (kW), which measures the model's output change in reaction to an input variation, namely an increase in price.

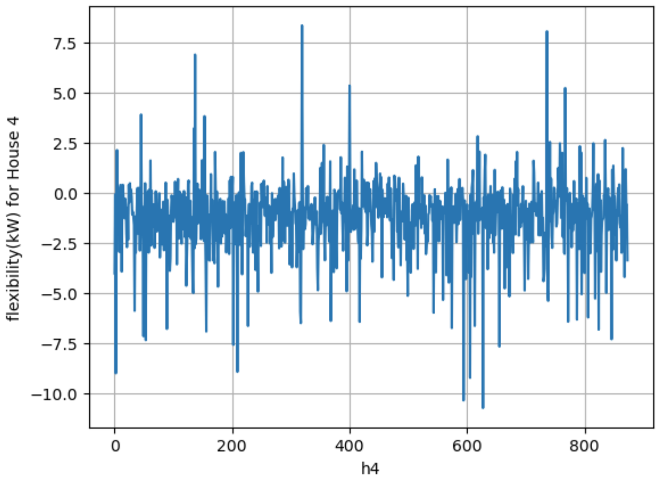

Figure 24. Flexibility for house 4 (h4)

Flexibility test with increased prices by 20 units:

Apply the provided data set to execute the model. This graphic demonstrates the adaptability of an LSTM network model in its response to variations in input data. The axis indicates the index of the dataset, which can refer to either time steps or individual data points within a series. Each index represents a distinct occurrence in which the model input was altered and a prediction was generated.

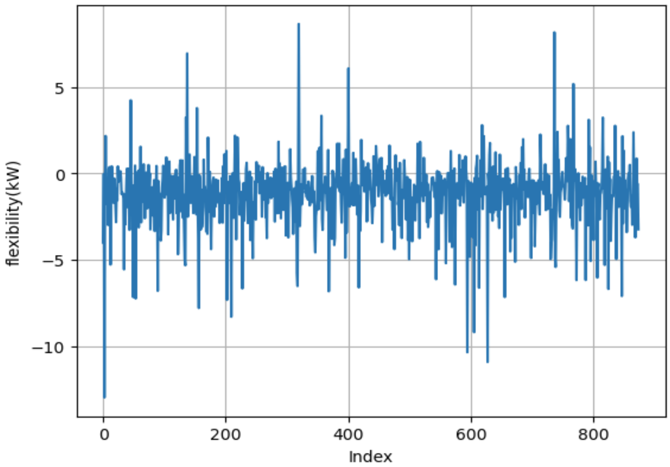

Figure 25. Flexibility curve based on data

The flexibility is measured in kilowatts (kW), indicating the scale of change in the model's output due to changes in the input.

Flexibility test for different house:

In order to assess the adaptability of various houses, we can examine the data of two specific houses: house 2 (h2) and house 4 (h4). Figure 26 depicts the flexibility values for house 2 when the price is increased by 20 units compared to the previous price. The X-axis, denoted as h2, shows the index of data points in the dataset. Every point on the axis represents a particular event in the dataset when the model was given an input (most likely the price) and generated a forecast. The Y-axis depicts the flexibility, measured in kilowatts (kW), which measures the model's output change in reaction to an input variation, namely an increase in price.

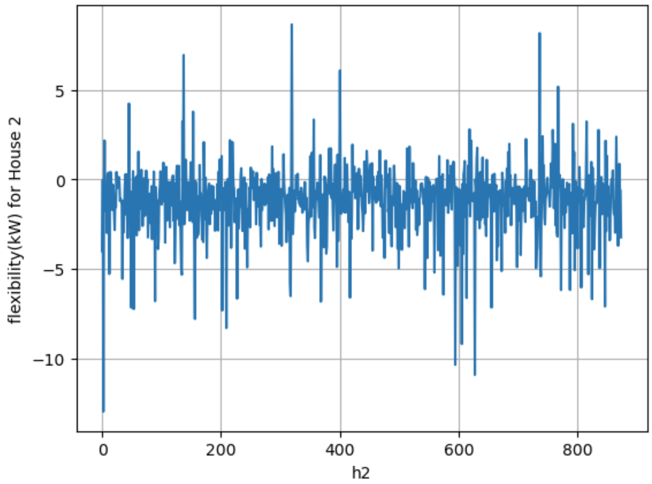

Figure 26. Flexibility for house 2 (h2)

This figure.27 illustrates that the flexibility values for house 4 when the price is 20 unit more than the previous one.

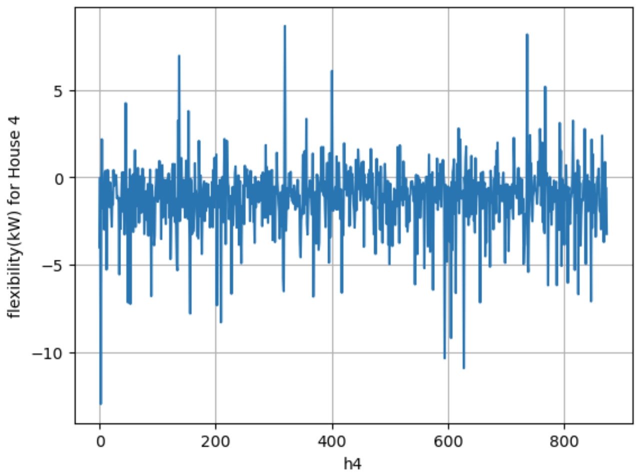

Figure 27. Flexibility for house 4 (h4)

Sensitivity to price changes:

In this scenario, the flexibility values fluctuate both above and below zero. The positive values on the flexibility plot indicate that there is a positive difference between the initial prediction and the changed prediction. In the above example, it is evident that increasing the input variable (price) leads to a decline in the forecasted outcome (demand) of the model. A positive flexibility number signifies an inverse relationship between price and demand, where an increase in price leads to a drop-in demand. This feature demonstrates a common economic phenomenon known as the inverse relationship between price and consumption, where people decrease their consumption as prices rise.

Similarly, negative numbers imply that the initial forecast is lower than the prediction made using the updated input. This is a consequence of the model producing greater values as the input is raised. This demonstrates a negative correlation, where the demand (output) increases as the price (input) increases. This refers to a common economic phenomenon in which people tend to increase their consumption as prices rise. In traditional market dynamics, this may seem contradictory unless the desired outcome is something that benefits from higher prices, such as revenue or an increase in the use of alternative resources when the primary resource gets too expensive.

Furthermore, the distribution of values around zero in both plots indicates an absence of any noticeable inclination towards positive or negative values. This suggests that the model does not exhibit a consistent tendency to overestimate or underestimate when price inputs are modified, without any noticeable distortion. According to the flexibility plots, the model consistently responds to changes in input across multiple houses. However, the specific changes in output can vary, possibly due to variances in the underlying data characteristics or unique aspects of each house.

House 2 has slightly more prominent extreme values with more frequency compared to House 4. This suggests that House 2 may exhibit a slightly higher degree of adaptability or responsiveness to changes in input, in comparison to House 4. The plot exhibits substantial fluctuation with multiple rises, encompassing both positive and negative values, indicating that the model's output for House 2 is very responsive to alterations in input values. A greater prevalence of more prominent extreme values (both positive and negative) indicates a higher level of responsiveness or adaptability to changes in input.

5.3 Analysis and interpretation

Flexibility plots are commonly employed to evaluate the response of a model's output, such as energy demand, to variations in input factors, such as prices. The flexibility measure is essential for comprehending the susceptibility of a system or model to external alterations. In this section mainly compare the flexibility of the consumer houses based on the three models.

5.3.1 Comparison of models

In order to determine the flexibility of the house, many models are employed in order to determine which house exhibits greater flexibility when the price is increased. Three distinct models are utilised in this context, namely RNN, GRU, and LSTM.

a) Recurrent neural network (RNN) model

The plots generated by this model exhibit oscillations around the zero line, however with lower frequency and amplitude of peaks compared to the GRU model. This suggests that the model has a more stable response, but it may have limited ability to respond to significant or abrupt changes in input. The fundamental recurrent neural network has the most consistent output with minimal variability in adaptability. This suggests a reduced susceptibility to alterations in input, which could be advantageous in situations that need constant performance and do not require spectacular reactions to input changes. The RNN model is appropriate for less complex tasks that demand system stability and less remarkable responses.

b) The gated recurrent unit (GRU) model

The graphs generated by this model exhibit a more tightly clustered distribution of fluctuations centred around the zero line. This indicates superior generalisation and a reduced tendency to respond excessively to perturbations in input, as compared to the LSTM model. The GRU appears to achieve a harmonious equilibrium between sensitivity and stability. The reaction of the LSTM is more intense compared to the other model, suggesting that it may have superior generalisation ability. This enhancement could potentially enhance the suitability of the GRU model for applications that require stability when dealing with diverse inputs.

The GRU model is likely the optimal choice overall, as it offers a harmonious combination of adaptability and reliability. The model captures a sufficient amount of input variability for most practical applications, without the extremes of the LSTM. This presumably enhances its robustness across a broader variety of situations.

c) The Long Short-Term Memory (LSTM) model

The graphs generated by this model have more prominent peaks and deeper troughs, indicating a heightened sensitivity to changes in the input. The extensive range of flexibility suggests that the model has the ability to accurately represent substantial variations in the input data, but it may also signal a risk of overfitting or excessive variability. The LSTM has the greatest sensitivity and variability, making it suitable in situations where catching significant input changes is crucial. Nevertheless, this may not always be preferable since it can result in overfitting and reduced predictability when applied to new data.

The LSTM model is suitable when the application necessitates capturing a high degree of sensitivity to input changes and when complex patterns in lengthy sequence data need to be modelled. The optimal model choice is typically contingent upon the unique criteria of application, encompassing factors such as the characteristics of the data, the trade-off between reliability and responsiveness, and the impact of the model's predictions on decision-making procedures. For example, in a dynamic pricing model for energy, greater sensitivity could enhance the effectiveness of demand response mechanisms, but it could also result in higher unpredictability in energy usage forecasts, necessitating the implementation of stronger management strategies.

6 Methodology

The initial stage of the methodology involves collecting and organising data, followed by exploring the data. The following stage involves pre-processing the data for the machine learning, and the final step involves applying many machine learning models, which include Regression model, Ensemble model, LSTM, RNN, and GRU, to predict the flexibility of a house dependent on the electricity price and time horizon. Finally, different performance metrics are employed to compare the performance of the various machine learning algorithms and determine the most effective approach. In conclusion, based on the stages of the models' optimisation and tuning procedure, choose the model that best predicts the price of electricity (Abumohsen at el., 2023).

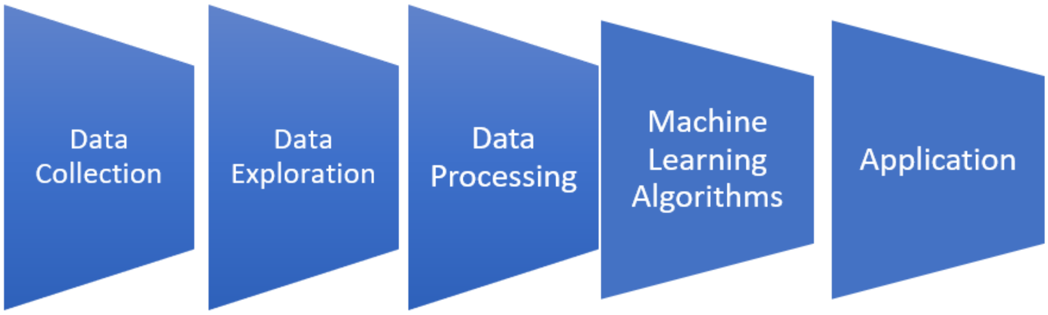

Figure 28. Basic workflow for forecasting models (Seyedzadeh at el.,2019)

6.1 Data collection and description

The data utilised in this thesis were acquired from the electrical consumption of seven residences in Vaasa. The electricity use of all houses was recorded on an hourly basis from January 1, 2021, to December 31, 2021, and stored in databases. The date is saved as an object in the form of a string. These seven household electricity consumption measures are recorded as float64 values. Float64 is a precise phrase that denotes a certain data type commonly employed in programming, namely in data manipulation and numerical computation libraries such as Pandas, which is utilised in the Python programming language. Float64 is an abbreviation for "floating point," which is a technique employed to represent real numbers that might have a fractional component. The "64" in float64 denotes that the data type uses 64 bits to store the information. For this particular scenario, a total of 64 bits are utilised. This floating-point encoding enables double precision, commonly referred to as such in other programming environments. It offers a broad range and substantial precision for numerical operations. The number representation has a range from around 10-308 to 10308 and provides a precision of 15 to 17 significant digits (‘IEEE Standard for Floating-Point Arithmetic - Redline’, 2019).

The dataset consists of 8758 rows and 10 columns, as depicted in Figure 29. This figure displays the first five records of the data collected from various households. The dataset includes the following main features: date hour, weekday, week number, month, year, temperature, holidays, and price (Abumohsen at el., 2023).

```
df=pd.read_csv("household2021.csv")
```

```
df.head(5)
```

	date	h1	h2	h3	h4	h5	h6	h7	Air temperature (degC)	Day-ahead Price [EUR/MWh]
0	2021-01-01 02:00:00	7.697	2.936	3.067	6.959	1.986	4.127	3.002	0.4	24.35
1	2021-01-01 03:00:00	5.661	3.933	3.090	5.268	2.010	4.140	1.391	0.6	23.98
2	2021-01-01 04:00:00	7.128	2.463	4.023	4.922	2.009	4.393	3.379	0.7	23.72
3	2021-01-01 05:00:00	3.593	2.009	3.073	4.851	2.050	1.826	1.725	0.2	23.73
4	2021-01-01 06:00:00	1.248	0.148	3.064	2.733	2.098	1.048	3.186	0.3	24.06

Figure 29. The initial five records in the dataset

6.2 Exploratory data analysis (EDA)

Exploratory Data Analysis (EDA) is a statistical methodology that investigates the existence of many concealed characteristics and patterns inside a dataset (Javed et al., 2021). All data related to electricity use were extracted and examined in this part. EDA analysis involves a thorough evaluation of the various characteristics, relationships, and concealed patterns included in electrical load data. In this study, many techniques were employed to evaluate and investigate the data, including autocorrelation, scatter plots, and correlation analysis using both Pearson and Spearman's correlation coefficients. Additionally, heat maps were utilised for data visualisation, as suggested by Barhak (2018). EDA plays a crucial role in the development of time-series forecasting models since it reveals and investigates the connection between input and output variables. EDA (Abumohsen at el., 2023) is characterised by its capacity to visually analyse and explore the connections between many variables, revealing previously undiscovered patterns.

6.2.1 Autocorrelation analysis

Autocorrelation (AC) is a statistical tool that examines the connection between the present values of a variable and its previous values. AC is a useful tool for detecting recurring trend patterns in data across time, including seasonality (such as daily, weekly, monthly, and yearly patterns), trends, and cycles by using the given in equation 29 (Flores et al., 2012), compute the autocorrelation of the time series data, specifically focusing on correlations that exceed 40 lags.

$$r_k = \frac{\sum_{t=k+1}^{n}(X_t-\bar{X})(X_{t-k}-\bar{X})}{\sum_{t=1}^{n}(X_t-\bar{X})^2} \tag{29}$$

Here,

r_k is the autocorrelation coefficient at lag k.

X_t is the value of the time series at time t.

$\bar{X}$ is the mean (average) of the time series over the entire period observed

n is the total number of observations in the time series

k is the leg, which is 40.

Figure 30. illustrates the auto-correlation of the electricity use. The variable "sum" from the dataset exhibits a correlation with its own previous values over an interval of 40-time delays. Within this particular context, the term "one lag" denotes a single instance of moving one-time step backwards in the stream of data. Here, a single lag represents a delayed version of a 1-hour time unit pushed backwards. Therefore, leg 40 signifies the

correlation with its value 40 units of time in the past. The electricity usage data is scaled to a range of values between 0 and 1. The plots indicate a significant correlation between the current electricity consumption and past lagged electricity consumption values, such as the electricity consumption from the previous hour and the electricity consumption at the same hour of the previous day (Javed et al., 2021). The y-axis represents the autocorrelation coefficient, which spans from -1 to 1. A coefficient around 1 indicates a robust positive correlation, indicating that as one data point increases, the preceding data points also tend to increase. A coefficient around -1 indicates a significant negative association. A value close to 0 indicates a minimal or negligible association.

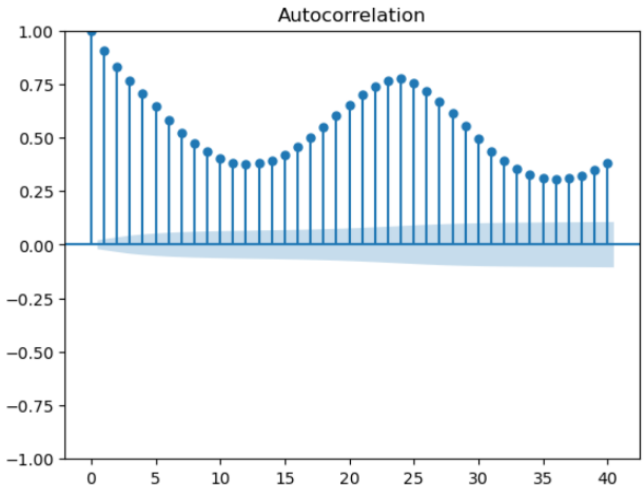

Figure 30. Autocorrelation for time series data

6.2.2 Correlation analysis

In statistics, the term "correlation" is used to describe the potential linear relationship between two continuous variables (Mukaka, 2012). Correlation can be used to predict one variable based on another. The rationale for employing correlation to choose features is based on the notion that valuable variables will have a robust correlation with the outcome. The heat map facilitates in evaluating the correlation coefficient between the features to determine if there is a sufficient relationship to build a deep learning model for predicting electrical consumption based on different features within the dataset. The heat map was generated using the python matplotlib and seaborn libraries, which compute the correlation coefficient (r) between the components using in equation 30 (Mukaka, 2012).

$$\boldsymbol{r} = \frac{\sum(\boldsymbol{x}-\overline{\boldsymbol{x}})(\boldsymbol{y}-\overline{\boldsymbol{y}})}{\sqrt{(\sum(\boldsymbol{x}-\overline{\boldsymbol{x}})^2 \sum(\boldsymbol{y}-\overline{\boldsymbol{y}})^2}} \tag{30}$$

Where,

r is the correlation coefficient

x and y are the two variables

$\bar{x}$ and $\bar{y}$ are the mean of x and y, respectively

Figure 31 demonstrates the connection among the features in the dataset. It reveals positive links between certain aspects, such as day-ahead price with h4, h6, and day-ahead price with month, hour. Additionally, it indicates negative relationships between air-temperature with h1, h3, and h7. Pleil et al. (2011) state that a heatmap is an effective tool for quickly identifying the variables with the strongest correlation, which may be used to explain or predict each other in a statistical model.

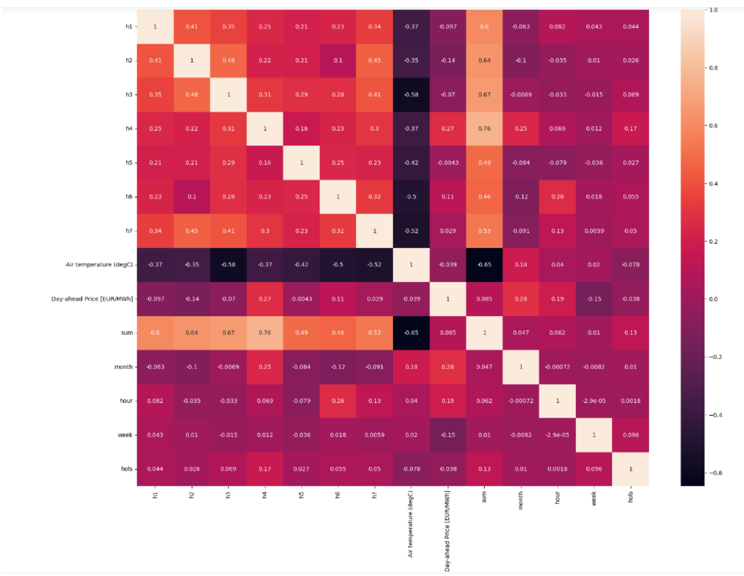

Figure 31. Correlation coefficient for each feature in the dataset

6.2.3 Scatter plots

A scatter plot is the most common kind of diagram used to visually depict the relationship between two variables (Birant et al.,2022). In order to identify the residential properties with the highest or lowest consumption of electricity, we will analyse the correlation between air temperature and the Day-ahead Price in Euros per Megawatt-hour. Figure 5-6 displays the power consumption of various houses, represented by a scatterplot, as it relates to both the air temperature and the day-ahead pricing. This visualisation aids in predicting which dwelling exhibits greater flexibility in terms of air temperature and day-ahead price.

Figure 32. demonstrates a correlation between temperature and the increase in variable h2. As temperatures rises from around -20°C to around 10°C, h2 also increases. However, beyond this point, h2 either levels off or slightly decreases. The point density is highest within the temperature range of 0°C to 20°C, and the variability in h2 diminishes as the temperature rises. This suggests that h2 exhibits more stability or attains a maximum value at elevated temperatures. Conversely, this house is more responsive to temperature increases and can also decrease electricity consumption in response to higher electricity prices.

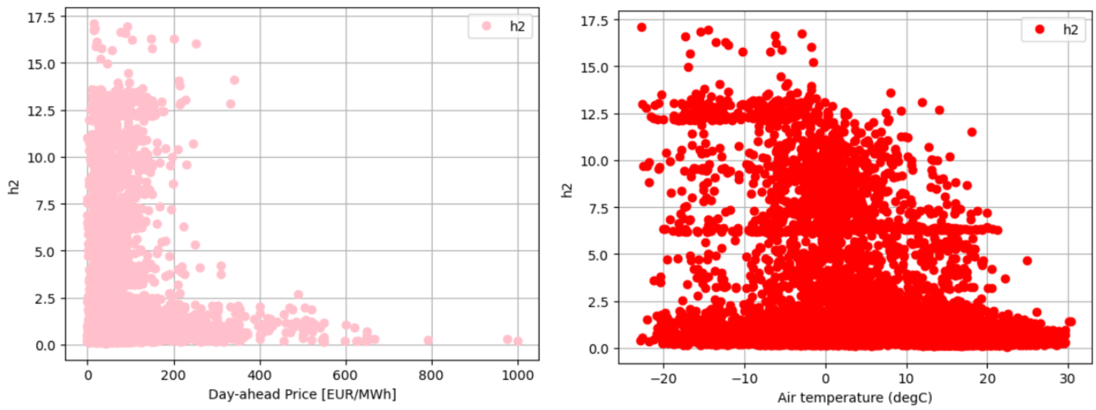

Figure 32. Electricity consumption for house 2 (h2).

Figure 33 indicates that as temperatures rise from cold to moderate, the basic response of home 4 appears to be consistently stable. Additionally, the consumption of power in this house shows relatively low sensitivity to changes in energy pricing.

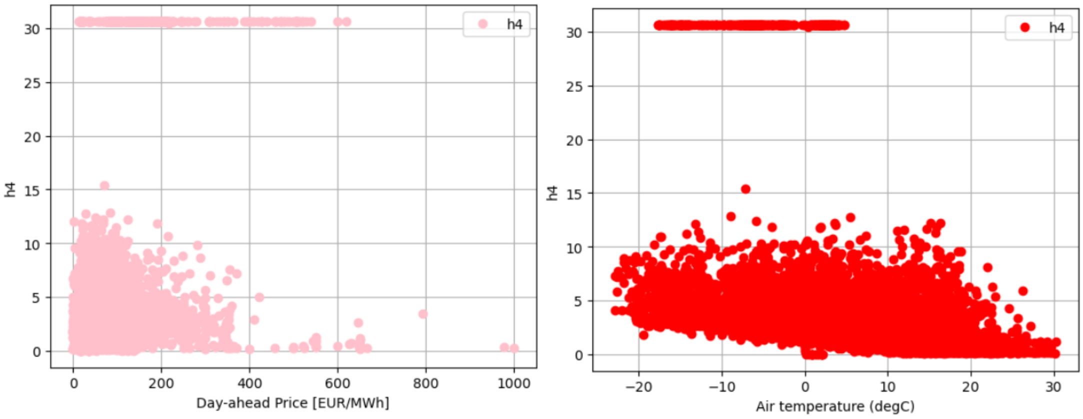

Figure 33. Electricity consumption for house 4 (h4).

6.3 Forecasting methodology

This section outlines the suggested methodology for predicting electricity prices using deep learning algorithms on a real dataset. The process begins with creating the dataset and explaining the pre-processing procedures, such as normalisation and feature selection. Next, the process involves constructing and training various comprehensive Machine learning (ML) models, such as different types of regression models, ensemble models, LSTM, RNN, and GRU, to predict electricity prices using historical data. Subsequently, hyperparameter tuning was conducted for LSTM, RNN, and GRU machine learning models. This involved optimising parameters such as the optimizer, activation function, learning rate, number of epochs, batch size, and number of hidden layers. The accuracy of each model was evaluated using the performance metric of root mean square error (RMSE). Ultimately, choose the most optimal forecasting model. The models in this study were developed using Python 3 and relied on the Pandas, Seaborn, Sklearn, Matplotlib, TensorFlow, and NumPy libraries (Abumohsen at el., 2023).

6.3.1 Data Pre-processing

Data pre-processing is an important stage in training machine learning models to utilise the optimal data structure. Without pre-processed data, the effectiveness of machine learning models may be compromised, leading to inadequate outcomes. Data normalisation and correlation feature processing are employed for data pre-processing, depending on the characteristics of the initial data (Abumohsen at el., 2023).

6.3.2 Feature selection

Feature selection is a method used to identify and measure the impact of one or more factors in a given dataset on the dependent variable. This study primarily uses the Auto correlation, a correlation matrix of two or more variables, and correlation coefficients to determine the relationship between one variable and other variables (Bampoulas at el.,2021).

Table 1 Summarises features considered in this study.

Feature Category	**Features**
Weather variables	Air temperature
Calendar information	Workday type (Weekend, weekdays, holidays), month, hours,
Historical data	Electricity consumption from house h1 to h7.
Miscellaneous	Sum, Day-ahead price
Autocorrelation function (ACF)	Select the N most dominant lag terms by utilising the ACF, Here 24 (legs) past values

6.4 Model selection

Depending on our dataset, features and problem types here different types of ML algorithm are most suitable to find out the flexibility of different houses in response of price.

6.4.1 Model shortlisting

In order to forecast the flexibility for various residences based on the price of electricity throughout a range of time periods, fourteen distinct models are employed to analyse the problem. Regression models such as linear regression, lasso, Ridge, Elastic net, and polylinear regression. Some examples of ensemble models include the Random Forest Regressor, Gradient Boosting Regressor, Extra Trees Regressor, XGBRegressor, LGBM Regressor, and KNeighbors Regressor. In this study, neural networks such as recurrent neural networks (RNN), gated neural networks (GRU), and long short-term memory (LSTM) are utilised.

6.4.2 Performance metrics selection

One of the most important steps after developing a model is to evaluate its training and predictive performance (Pham at el., 2020). To determine the performance of the model root mean square error (RMSE) metric is used to evaluate and compare models. RMSE calculated the quantitative comparison of the correspondence between the predicted and observed data by using equation 31 (Bampoulas at el., 2021).

$$\text{RMSE}=\sqrt{\sum_{i=1}^{N}\frac{(Y_i-\hat{Y}_i)^2}{N}} \tag{31}$$

Here, Y_i is the actual value

$\hat{Y}_i$ is the predicted value

N represents the test set size (874)

6.5 Model training

Develop the model using selected features and algorithms to learn from the data. This figure 34. illustrates the general overview of machine learning model working procedure.

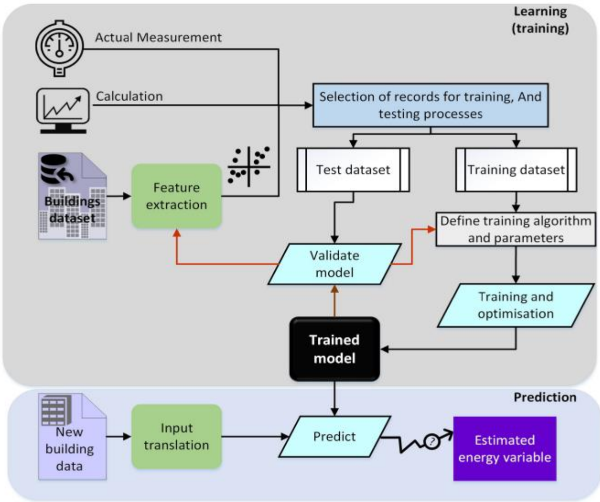

Figure 34. General process of Machine learning model (Seyedzadeh at el.,2019)

6.5.1 Training and test set determination

The test set size is therefore determined by the model prediction horizon. The optimal training set size is determined parametrically to avoid underfitting and to keep the prediction model updated to the most recent occupancy and weather conditions. In this work, 90 % data is used for training purpose and 10% data is used for testing purpose.

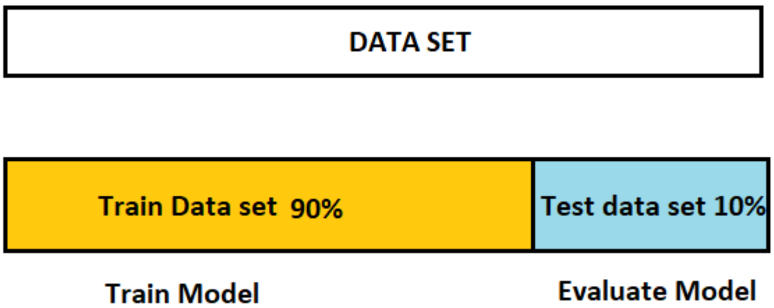

Figure 35. General diagram of splitting data set

6.5.2 Regression model training

The training of the regression model involves using the variables 'month', 'week', 'hour', 'hols', 'temperature', and 'price' from the dataframe df_train as the variables (X_train), and the variable 'sum' from the same dataframe as the target variable (y_train). This code extract selects specific columns from the dataframe df_train to build the feature set, which is stored in the variable X_train, and the target variable, which is stored in the variable y_train. The features encompass the time units of 'month', 'week', and 'hour', as well as the variables 'hols' (holidays), 'temperature', and 'price'. The dependent variable is 'sum'.

X_test and y_test is assigned the values of the 'month', 'week', 'hour', 'hols', 'temperature', and 'price' columns from the df_test dataframe, and the 'sum' column from the df_test dataframe, respectively. Similarly, this function retrieves the identical set of characteristics and the desired outcome from the testing dataset (df_test). The attributes X_train.shape and X_test.shape should display the count of observations and the count of features. Here, X_train consists of 7861 observations with 6 characteristics, while X_test consists of 874 observations with the same 6 features.

The parameters y_train.shape and y_test.shape indicate the quantity of observations in the target dataset. The target variable is represented by a single column. The shapes of the target values are (7861,) for the training set and (874,) for the testing set. This indicates that there are 7861 target values for training and 874 target values for testing. The division between training and testing is beneficial for assessing the model's performance on unfamiliar data.

Data is crucial in the training process of a regression model since it determines the link between the independent variables (features) and the dependent variable (goal). In addition, a regression model learns by iteratively altering its parameters in order to minimise a loss function, commonly the Root Mean Squared Error (RMSE), which is used in regression tasks. The objective is to identify the line (or hyperplane in many dimensions) that provides the most accurate fit to the data by minimising the discrepancy between the anticipated values and the actual values. The model will utilise the training data to establish a line that can reliably forecast the target variable based on the given features.

The model computes the coefficients for each feature, which indicate the magnitude and nature (positive or negative) of the association between each feature and the target (Pandey et al.,2022).

6.5.3 Ensemble model training

Each ensemble model, such as Random Forest Regressor, Gradient Boosting Regressor, Extra Trees Regressor, XGBRegressor, LGBM Regressor, and KNeighbors Regressor, is fitted to the training data (X_train, y_train). The Root Mean Square Error (RMSE) is then calculated for both the training and testing sets.

Here, X_train consists of 7861 observations with 6 characteristics, while X_test consists of 874 observations with the same 6 features. Similarly, this process retrieves the identical set of characteristics and the desired outcome from the testing dataset (df_test). The features encompass the time units of 'month', 'week', and 'hour', as well as the variables 'hols' (holidays), 'temperature', and 'price'. The dependent variable is 'sum'. The ensemble models are subsequently trained on the dataset using the model.fit(X_train, y_train) method. The function get_rmse is used to determine the root mean square error (RMSE) for both the training and test datasets. This function performs prediction and error computation, and it outputs the root mean square error (RMSE), which measures the average magnitude of the prediction error. Figure 36 depicts the overall methodology for determining the target variables.

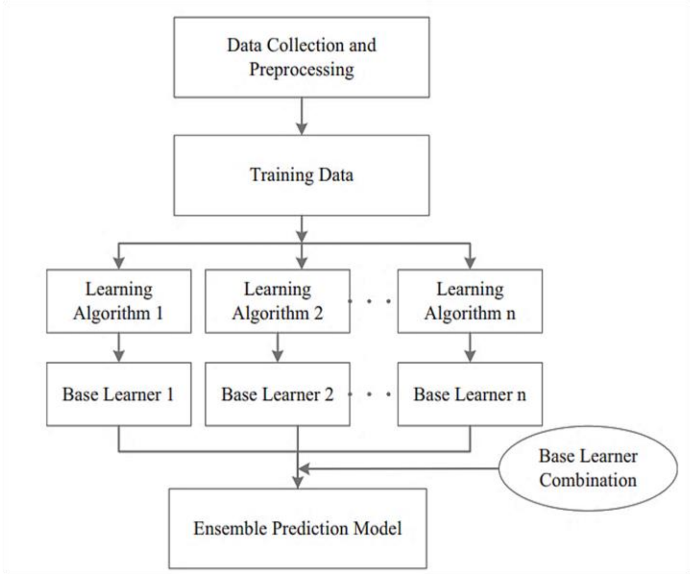

Figure 36. General framework for ensemble model (Wang & Srinivasan.,2017)

6.5.4 Neural network model training

Three distinct neural network models have been implemented, namely recurrent neural network (RNN), gated neural network (GRU), and Long short-term memory (LSTM). In this work, 7861 training data, 874 testing data, and 26 characteristics were utilised to execute the models.

The data consists of 7861 observations with 26 features in the X_train set, and 874 observations with the same 26 characteristics in the X_test set. Similarly, this function retrieves the identical set of characteristics and the desired outcome from the testing dataset (df_test). The characteristics consist of the following: 'month', 'week', 'hour', 'hols', 'temperature', 'price', and 24 legs. The dependent variable is 'sum'.

a) Recurrent Neural Network (RNN)

Recurrent neural networks (RNNs) were specifically developed to assess data that occurs in a sequence across time, and have been effectively utilised in various fields. A Recurrent Neural Network (RNN) analyses incoming sequence or time-series data by examining individual vectors at each step, while maintaining the hidden information from prior time steps. At each specific moment, the output was returned to the network in order to improve its result (Abumohsen at el., 2023).

The input data is entered into the recurrent neural network (RNN), passing through each layer. The Recurrent Neural Network (RNN) employs its internal state, also known as the hidden state, along with the current input to calculate the output. Once the final output is generated, the error is calculated by implementing the loss function. Additionally, the training of RNNs is essentially sequential and requires meticulous handling of memory states that store information about preceding data points in a series. RNNs are very suitable for tasks such as time series forecasting and natural language processing.

b) Gated neural network (GRU)

A gated neural network (GRU) is a specific variant of a recurrent neural network. This model excels at handling the long-term dependencies in data. This refers to the process of acquiring knowledge from data, when the present outcome is influenced by data points that were met in the past, even if they occurred a long time ago in the sequence. The primary issue with the RNN model was this. The GRU model consists of two gates: an update gate and a reset gate. The update gate in this paradigm sets the proportion of previous information that should be transferred to the future. This gate primarily concerns the amount of previous information that needs to be retained for the next prediction. The reset function determines how much irrelevant information from the past needs to be forgotten, as it can disrupt the sequence. In addition to these two gates, this model features a concealed stage that primarily chooses which information to retain (update) and which information to disregard based on the outputs of the gates (Mahjoub et al.,2022).

In order to execute the model, the GRU receives input from the current time step in the sequence. It merges the latest input with the previous concealed state and determines the extent to which the previous memories should be retained. During the training process, the output and hidden state of each step are computed in a sequential manner. The update and reset gates adapt their behaviour according on the discrepancy between the output and the anticipated outcome (Wang el at., 2020).

c) Long short-term memory (LSTM)

Long Short-Term Memory networks (LSTMs) are a specific type of Recurrent Neural Network (RNN) that have the ability to learn and capture long-term relationships within sequences of data. This is especially beneficial for applications like time series forecasting, where historical context from a significant distance in the sequence is crucial for generating accurate forecasts. LSTMs were specifically developed to address the issue of "vanishing" and "exploding" gradients that can arise in conventional RNNs (Sagheer & Kotb, 2019). LSTMs address this issue by incorporating a component known as a "cell," which is capable of retaining information in memory over extended durations. An LSTM cell comprises three primary components that govern the information flow. The three primary constituents consist of the forget gate, input gate, and output gate.

The forget gate determines which information should be eliminated from the cell state. The model examines the preceding concealed state ht-1 and the current input Xt, and generates a value ranging from 0 to 1 for each number in the cell state C t-1. In this context, the value of 1 signifies the action of "completely keeping" something, whereas the value of 0 signifies the action of "completely getting rid of" something. Conversely, the input gate determines the specific information that will be stored in the cell state (Wang et al., 2020). Within the input gate, a tanh layer is used to generate a new vector of candidate values. This vector is then merged with the existing cell state to provide the updated cell state, denoted as Ct. The output gate filters the new cell state, denoted as Ct. This gate determines which components of the cell state are included in the output, which then becomes the new hidden state ht. The hidden state is employed to produce the ultimate outcome of the LSTM unit, which may take the form of a forecast, such as the subsequent value in a time series (Mahjoub et al., 2022).

d) Hyperparameters tuning for neural network models

This section presents the hyperparameters employed in this research to achieve optimal outcomes for the applied models. In this section, we will explore the optimal parameters that dictate the structure of models used for predicting electrical loads. This process is commonly referred to as hyperparameter tuning (Veloso al et., 2021). Tuning refers to the procedure of selecting the most favourable combination of hyperparameters for the learning algorithm (Abumohsen at el., 2023).

The parameters used in this research are listed as follows:

- Best optimizer: For this three model Adam optimizer is used.
- Activation function: For this three-model linear activation function are used but in GRU and LSTM also used relu activation function.
- Learning rate: Learning rate is 0.001 for those three models.
- The number of epochs: In RNN model number of epochs is 60 and for other two model number of epochs is 50.
- Batch size: In RNN model, batch_size is 62 and for other two model batch_size is 32.
- The number of hidden layers: In RNN 48 hidden layers are used. GRU and LSTM used 32 hidden layers are used.

6.6 Forecasting mechanisms of different model

6.6.1 Data set information for neural network

Our data set are splits into two part such as training fs from the training and testing data frames for predicting the target variables. The selected features and the target variable are then concatenated along the axis 1 using pd.concat to create X_train and X_test. The data is seperated into two parts such as features (stored in X_train and X_test) and target variables (stored in y_train and y_test).

```
X_train = pd.concat([df_train.iloc[:,15:], df_train[['temperature','price']]], axis=1)
```

```
X_train.shape
```

```
(7861, 26)
```

```
X_test =  pd.concat([df_test.iloc[:,15:], df_test[['temperature','price']]], axis=1)
```

```
X_test.shape
```

```
(874, 26)
```

```
y_train.shape
```

```
(7861,)
```

```
y_test.shape
```

```
(874,)
```

Figure 37. Data set information for modelling

Here, X_train = pd.concat([df_train.iloc[:,15:], df_train[['temperature','price']]], axis=1) indicates that we selects all columns from the 15th (index 14) onwards and also temperature and price as a features. The pd.concat function combines these data-frames along axis 1 (columns), resulting in a new data-frame X_train that contains both the selected features and the target variable. Here, X_train.shape (7861,26) indicating 7861 number of samples in each testing set and 26 features.

On the other hand, for testing data we select all columns from the 15th (index 14) onwards and also temperature and price as a feature.

In addition, X_test = pd.concat([df_test.iloc[:,15:], df_test[['temperature','price']]], axis=1). Here, X_test.shape (874,26) indicates, we have 874 number of samples in each testing set and 26 features.

Furthermore, the structure of y_train.shape (7861,) indicates that there are 7861 rows, representing the sum of electricity uses information about for 7861 number of samples in our training dataset. Similarly, the structure of y_test.shape (874,) indicates that there are 874 rows, representing the sum of electricity uses information about for 874 number of samples in our training dataset. The model is trained using this data to determine the relationship between several features (such as price and temperature) and the sum of electricity consumption.

Now, each row in (X_train and X_test) likely represents a single feature vector. This vector contains the values of all the features (columns) for a specific training sample (electricity consumption for different houses in our dataset). During training, the model utilises the feature vectors from X_train as input. In addition to each feature vector, the model is provided with the associated target value from y_train.

6.6.2 Forecasting mechanisms of recurrent neural network

Recurrent neural networks (RNNs) are a type of neural network that performs tasks like time series forecasting, where the primary purpose is to predict future values based on previous value data. There are two main mechanisms that RNNs use for forecasting, such as backpropagation through time and long short-term memory. Backpropagation through time (BPTT) is a crucial technique for training recurrent neural networks (RNNs) to accurately predict future values. It works by propagating the error signal backward through the network's recurrent connections, allowing it to adjust its weights and improve its predictive performance. Another option is long short-term memory, which is a form of RNN that works particularly well with long-term data dependencies. Furthermore, a number of techniques are used to improve the forecasting of RNN, such as regularisation, dropout, and early stopping to prevent overfitting (Pra, 2020).

To build this RNN model, import the Sequential module from tensorflow.keras is used to build sequential models by adding layers in a sequential way. Additionally, import layer types including SimpleRNN, Dense, Embedding, and Dropout, as well as various optimizers including Adam and SDG.

In the specified RNN layer, the parameter "units" with a value of 48 indicates the number of hidden units that will be used to learn the patterns in the input. In addition to set the parameter "return_sequences" to True. By utilising this expression, the layer returns the complete output sequence, encompassing all elements rather than only the last one.

In the first layer of the RNN, the code input_shape=(X_train.shape[1], 1) indicates a collection of integers that defines the expected dimensions of the input data for the layer. Additionally, in input data set we include the number of features (columns) that represents the first dimension of the input data (number of features per sample) from our data set and in the second dimension of input data create a tuple that hold multiple items as like a list in this part we set 1, because our data is single value for each time step.

In the second layer my_rnn.add(SimpleRNN(units=48, return_sequences=True)) to process the output from the first layer and potentially learn more complex patterns. Moreover, in the third layer my_rnn.add(SimpleRNN(units=48, return_sequences=False)) meaning that it will only return the final output and also summarising the acquired information from the complete sequence.

The final layer of my_rnn.add(Dense(units=1, activation='linear')) illustrates that, dense layer with units 1 and linear activation function. Additionally, unit 1 specifies that the layer will consist of only one output unit. This implies that the model will be trained to combine the input obtained from the previous recurrent layers into a single output value that represents the prediction. On the other hand, linear activation function in the model indicates a linear relationship between the input features and the target variable (sum). The final prediction is generated by dense layer, which uses the patterns learned by the previous recurrent layers in order to process sequential data.

During the training process, we utilise the Adam optimizer with a learning rate of 0.001. This optimizer updates the weights of the model and the learning rate determines how weights are modified depending on the training errors. Moreover, mean_squared_error is used as a loss function to measure the difference between the model's predicted and the actual target values.

Furthermore, to configure the training process, we split our dataset into training data (X_train, y_train) and validation data (X_test, y_test) for the purpose of determining the model's performance. The function rnn_model.fit is used to train the model. It takes the input data X_train (7861, 26) and the corresponding target values y_train (7861,). The validation data, consisting of X_test (874, 26) and y_test (874,), is used to evaluate the model's performance during training. The training process is repeated for 60 epochs, with a batch size of 64. During each epoch, the complete training dataset is processed by the model, which undergoes training and modifies its internal parameters (weights and biases) to enhance its predictive capabilities. The first epoch model requires 8 seconds to finish. The average processing time for each training step (batch) throughout the period is 30 ms.

After completing the first epoch, the training loss is 155.5286 and the validation loss is 112.7503. Additionally, loss is a quantitative measure that evaluates the difference between the predictions and the actual target values included in the training data. A smaller loss number often signifies better outcomes on the training data. The validation data, consisting of X_test and y_test, is employed to evaluate the model's ability to generalise to new and unseen data. The validation loss, at 112.7503, is smaller than the training loss, indicating a favourable outcome. It indicates that the model may be exhibiting good performance and is not yet overfitting to the training data.

After 60 epochs, the training loss significantly decreased to 15.9437, suggesting that the model effectively learned from the training data and improved its ability to make predictions. Furthermore, the validation loss also decreased to 18.6923, suggesting a close resemblance to the training loss. This suggests that the model is demonstrating strong performance on both the training and validation data, indicating that it has successfully learned the underlying patterns. After performing 60 epochs, the root means square error (RMSE) values for the training and testing data are 3.8344 and 4.3235, respectively. The above information indicates an average difference between the expected outcomes and the actual target values. On the other hand, smaller RMSE values indicate higher performance of the model. The training error (3.8344) is marginally smaller than the testing error (4.3235) in this instance.

Consequently, the RNN model was trained by generating predictions on both the training and testing data, and the Root Mean Square Error (RMSE) was calculated for each dataset. The RMSE values offer valuable insights into the model's ability to generalise to unfamiliar data.

6.6.3 Forecasting mechanisms of long short-term memory (LSTM)

Long Short-Term Memory (LSTM) is a specialized recurrent neural network (RNN) specifically designed for processing sequential data, particularly time series information. LSTM architecture is fundamentally composed of memory cells, which are specifically designed to retain and access information over extended time intervals (Pra, 2020).
To build this LSTM model, import the Sequential module from tensorflow.keras is used to build sequential models by adding layers in a sequential way. We basically used first, second and dense layer to run the model.

In the first layer of the code is my_LSTM.add(LSTM(units=32, return_sequences=True, activation='relu', input_shape=input_shape)). The units 32 indicates the hidden units in the LSTM layer. The parameter return_sequences=True determines whether the layer will return the full sequence of hidden states or simply the final output for each input series. When set to true, the layer will return the complete sequence of hidden states, which may then be used as input for the following LSTM layer. The relu activation function is utilised in this layer. Rectified Linear Unit (ReLU) activation provides an effective way to solve vanishing gradients, introduce non-linearity appropriate for Long Short-Term Memory (LSTM) systems, and balance computing efficiency. The input_shape parameter in the model indicates the structure or dimensions of the data. Also, in the second layer, the parameter return_sequences=false determines the layer will only return the final output for each input sequence. A dense layer refers to a completely connected layer inside a model. The code for the dense layer is "my_LSTM".Include a dense layer with 1 unit and a linear activation function. The dense layer with a unit of 1 determines the number of output units in this layer, as we are forecasting a single value. This layer uses linear activation functions, which cause the output to be a linear function of the inputs.

In addition, the line of code "lstm_model = my_LSTM((X_train.shape[1], 1))" determines the input shape for the model based on the training data X_train. X_train.shape[1] indicates the number of features per timestep in the training data, while 1 refers to the number of channels in the data (one feature per timestep).

During the training process, we utilise the Adam optimizer with a learning rate of 0.001. This optimizer updates the weights of the model and the learning rate determines how weights are modified depending on the training errors. Moreover, mean_squared_error is used as a loss function to measure the difference between the model's predicted and the actual target values. To train the model final code is lstm_model.fit (x=X_train, y=y_train, validation_data= (X_test, y_test), epochs=50, batch_size=32). Here, split our data set x=X_train array contains the features for each timesteps and y=y_train array contains the actual values model will predict. In validation_data = (X_test, y_test) monitor the model performance during training. In this case, the model is trained for 50 epochs, which indicates that the complete training dataset will be run through the network 50 times for training purposes. In addition, it utilises 32 batches, where a batch refers to a subset of our training data used for a single update throughout the training process. Following table 2 showed, in the data processing the first epoch model requires 12 seconds to finish. The average processing time for each training step (batch) throughout the period is 30 ms.

After completing the first epoch, the training loss is 72.7003 and the validation loss is 36.8400. Additionally, loss is a quantitative measure that evaluates the difference between the predictions and the actual target values included in the training data. A smaller loss number often signifies better outcomes on the training data. The validation data, consisting of X_test and y_test, is employed to evaluate the model's ability to generalise to new and unseen data. The validation loss, at 36.8400, is smaller than the training loss, indicating a favourable outcome. It indicates that the model may be exhibiting good performance and is not yet overfitting to the training data.

Table 2 Epoch information in data processing

Epoch	Processing time	Training loss	Validation loss
1	12s 30ms/step	72.7003	36.8400
2	7s 29ms/step	36.2526	30.0340
50	7s 28ms/step	15.7597	19.7352

On the other hand, in the second epoch requires 7s to finish. The average processing time for each training step (batch) throughout the period is 29 ms. After completing the second epoch, the training loss is 36.2526 and the validation loss is 30.0340.

After 50 epochs, the training loss significantly decreased to 15.7597, suggesting that the model effectively learned from the training data and improved its ability to make predictions. Furthermore, the validation loss also decreased to 19.7352, suggesting a close resemblance to the training loss. This suggests that the model is demonstrating strong performance on both the training and validation data, indicating that it has successfully learned the underlying patterns. After performing 50 epochs, the root means square error (RMSE) values for the training and testing data are and 3.6880 and 4.3540 respectively. The above information indicates an average difference between the expected outcomes and the actual target values. On the other hand, smaller RMSE values indicate higher performance of the model. The training error (3.6880) is marginally smaller than the testing error (4.3540) in this instance.

Consequently, the long short-term memory (LSTM) model was trained by generating predictions on both the training and testing data, and the Root Mean Square Error (RMSE) was calculated for each dataset. The RMSE values offer valuable insights into the model's ability to generalise to unseen data.

6.6.4 Data set information for ensemble model

In order to execute the model, we divide the data-frame into two separate parts: one for training and the other for testing. Here, 10% of the data is allocated for testing reasons, while the other 90% is allocated for training purposes. In this data set X_train and X_test contain the features used to train and test the model and the fratures are month, week, hour, hols (holidays), temperature and price. On the other hand, y_train and y_test contain the target variable, which is sum.

Additionally, X_train.shape: (7861, 6) indicates that in training features dataset has 7861 rows and 6 columns. Each row corresponds to a different data point (or example), and each column represents a feature used for training the model. The features, as mentioned earlier, are month, week, hour, hols (holidays), temperature and price.
Moreover, y_train.shape: (7861,), demonstrates that the training labels dataset consists of 7861 rows and is represented as a one-dimensional array. Every data point in this dataset corresponds to the target variable (sum) for all 7,861 data contained in the training dataset.

Table 3 Data set information for features and target value

Data set	Features	Target Value
X_train.shape (7861,6)	Month, week, hour, hols (holidays), temperature and price	
y_train.shape (7861,)		Sum
X_test.shape (874, 6)	Month, week, hour, hols (holidays), temperature and price	
y_train.shape (874,)		Sum

On the other hand, X_test. shape: (874, 6), indicates that in testing features dataset has 874 rows and 6 columns. Each row corresponds to a different data point (or example), and each column represents a feature used for training the model. The features, as mentioned earlier, are month, week, hour, hols (holidays), temperature and price.
Moreover, y_train.shape: (874,), demonstrates that the testing labels dataset consists of 874 rows and is represented as a one-dimensional array. Every data point in this dataset corresponds to the target variable (sum) for all 874 data contained in the testing dataset.

6.6.5 Forecasting mechanisms of gradient boosting regression

Gradient-boosting regression uses a forward stage-wise method for forecasting. In every stage, a new regression tree is fitted based on the residual errors from earlier model estimations. Then the tree's predictions are added to the overall forecast, and the process continues until the stopping requirement has been satisfied. The final prediction is the sum of the predictions from all regression trees, combining the strengths of many trees to provide a more accurate and robust forecast (Ves et al.,2019).

The first stage of this model involves training a basic decision tree using the provided training data (X_train, y_train) and the features (X_train) to make predictions for the target variable (y_train). The decision tree calculates residuals by comparing the actual target values with the original prediction. Each tree is constructed individually, with a specific emphasis on understanding and addressing the errors (residuals) of the preceding tree. Furthermore, a new decision tree is trained utilising the residuals as the target variable and the features (X_train) as predictors. The predictions generated by this new tree are combined with the predictions from the previous trees in the ensemble. Consequently, the Root Mean Square Error (RMSE) is computed on the training data by using the adjusted ensemble predictions and the actual target values (y_train). The training RMSE value 4.67 indicates that the model is capturing the data patterns in an effective way and test RMSE value 4.78 is closely resembles the training RMSE, indicating a positive outcome. This indicates that the model is successfully adapting its learned patterns to new data (testing data) and is not only memorising the training data. The evidence illustrates that lower RMSE value indicates that model forecast the future value well.

6.6.6 Forecasting mechanisms of extreme gradient boosting regression (XGB Regression)

The XGB Regression model is trained on a dataset for predicting new data. In order to make a prediction, the model calculates the predictions of each tree individually. The final prediction is a combination of the predictions from all of the trees. This model uses various techniques for predicting new data. Some of the techniques are given below:

One of the common predictions techniques is Out-of-bag. This prediction technique is used for the data that are not used in the model. Its normally estimate the model's generalization error. The most effective technique is cross-validation. This technique is used for evaluating the performance of a model on unseen data. In this technique, data are divided into a number of folds and using each fold as a validation set. Early stopping is a regularization technique used in machine learning to reduce the overfitting. In addition, it monitors the performance of the model on a validation set and stop training if the performances of the model start to decrease (Houda et al.,2022).

The first stage of this model involves training a basic decision tree using the provided training data (X_train, y_train) and the features (X_train) to make predictions for the target variable (y_train). This decision tree using a subset of the training data to predict the target variable (Sum) based on the features(X-train). This model calculates the residuals by comparing the actual target values with the original prediction. Based on the previous residuals it creates new tree and every time new tree tries to correct the mistakes made by previous one until the stopping criterion is met (e.g., reaching a predefined number of trees or no significant improvement in performance). Once the training is complete, XGBoost uses the final ensemble of trees to predict the target variable on the unseen testing data (X_test). In addition, XGBoost computes the Root Mean Square Error (RMSE) on the complete training dataset (X_train, y_train) to evaluate the model's fit to the training data and identify any potential overfitting (excessive memorization of the training data).

Afterwards, the model calculates the Root Mean Square Error (RSME) using the testing data to assess the model's ability to generalise to new and unseen data. In conclusion, the root means square error (RMSE) for our training data is 2.45, whereas the RMSE for our testing data is 4.22. Furthermore, the lower training RMSE value suggests that the model is effectively fitting the training data. Furthermore, the testing value exhibits a little greater root mean square error (RMSE) compared to the training value, although it remains quite low. This indicates that the model exhibits satisfactory performance on the unseen data. In general, the RMSE values indicate that XGBoost is a reliable model for forecasting the target value.

7 Result and analysis

In this study, trained and evaluated ML algorithms based on the seven houses electricity consumption from 2020 to 2021 in Vaasa regions. The Primarily objective of this study is to revels the flexibility of houses based on different features and which ML model is more accurate based on the RMSE value to predicts the future consumption.

7.1 Flexibility of different customers

In the term of flexibility, individual customers play an increasingly essential role in the forecasting. Electricity consumption of a single customer is more dependent on the underlying human behaviour. For the individual customers, the number of outliers varies greatly from one household to another (Paterakis et al., 2017). In this study, most flexible consumers are described in chapter 5 based on the different neural network model.

To find-out the flexible consumer used three different ML models such as RNN, GRU and LSTM. In this study, based on the price value determine which house is more flexible with another house. Basically, based on the information of seven different houses used three different model with increasing the price from 10 to 20 and also observed the flexibility of houses. Flexibility basically determine when price increases consumer reduced their consumption or price decreases consumer increase their consumption.

In RNN model, house 4 shows more flexibility compared with other houses. On the other hand, based on the GRU and LSTM model, in both case house 2 shows more flexibility compared with other houses.

7.2 Forecasting model performance

First of all, according to the training and testing results of each model in the data set, seven houses electricity consumption from 2020 to 2021. It can be seen that the accuracy of the different machine learning method such as regression model, ensemble model and recurrent neural network (RNN), gated recurrent units (GRU) and Long short-term unit (LSTM) (Wang et al., 2020). To validate the models and compare them to each other in terms of training and validation performances, computed RMSE performance metrics processing both training and validation datasets (Pham et al., 2020).

7.2.1 Regression model

In traditional linear regression, it is assumed that there is a linear relationship between the features and the target variable. However, in real-world problems, many factors do not satisfy the linear assumption (Zhou et al., 2023).

In this study five regression models are used such as linear regression, lasso, ridge, elastic net and polylinear regressor. Accuracy is clearly the most significant criterion in the evaluation of the performance success of the prediction methods (Willmott & Matsuura.,2005). RMSE is used to measure the accuracy of the model. It measures of the average magnitude of the errors between predicted and actual observations, a lower RMSE value indicates a higher level of accuracy in prediction (Zhou et al., 2023).

Table 4 RMSE value of training and testing data for different regression model

Model name	**RMSE value on Training**	**RMSE value on Testing**
Linear Regression	8.22	7.98
Lasso	8.27	8.01
Ridge	8.22	7.98
Elastic Net	8.26	8.01
Poly Linear Regressor	4.61	6.44

On the other hand, higher value of RMSE indicates a lower level of accuracy in prediction. A high RMSE indicates the presence of outliers or large errors, prompting further investigation into model performance or potential data issues. Outliers, which are significantly different from other data points, can drastically increase the RMSE (Karandish & Šimůnek.,2016).

In this study, compere to the RMSE value of with each of the regression model it showed that training and testing RMSE value of poly liner regression is lower compared with others.

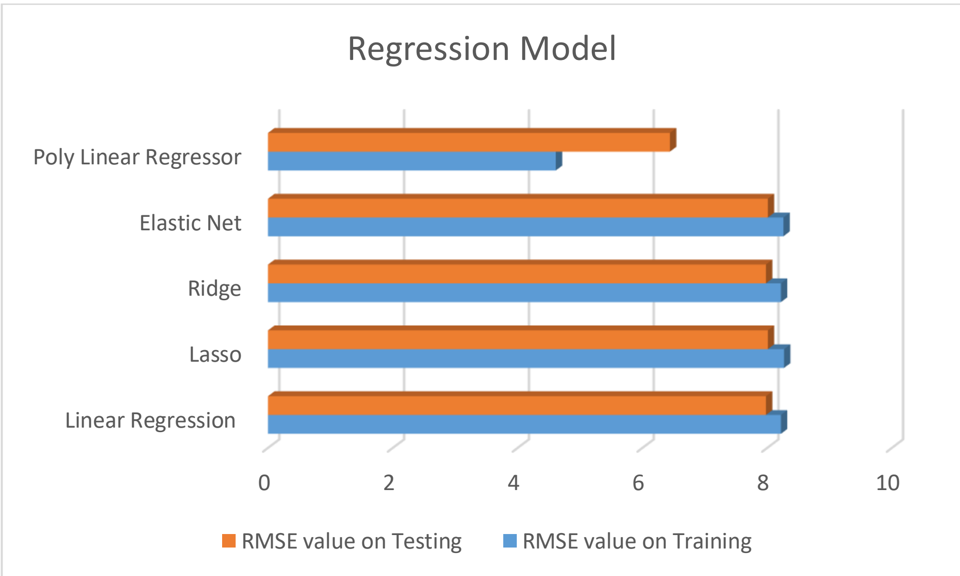

Figure 38. Different regression model analysis based on RMSE

The chart shows a comparative analysis of how each model contributes to the overall RMSE values in a dataset, which is a measure of how well each model predicts the dependent variable. From this figure it is showed that compare with other model, poly Linear regressor shows less of error where another model shows more error.

The Poly Linear Regressor fits a nonlinear relationship between the dependent and independent variables. It has the lowest RMSE on the training set, indicating it has learned the training data well. However, the increase in RMSE from training to testing (from 4.61

to 6.44) suggests some overfitting, where the model is too specifically tuned to the training data and does not generalize as effectively on the testing data.

Polynomial Linear Regressor can model non-linear relationships between the dependent and independent variables. On the other hand, linear models including Ridge, Lasso, and Elastic Net are build a liner relationship between the dependent and independent variables. If the dataset contains inherent non-linear patterns, a polynomial model is capable of capturing these complexities more effectively, leading to better performance. Despite the potential for overfitting, the Poly Linear Regressor achieved a significantly lower RMSE on the testing set than all linear models, indicating that it generalized better in this instance (Tyagi et al.,2022).

7.2.2 Ensemble model

In this study six ensemble models are used such as Random Forest Regressor, Gradient Boosting Regressor, Extra Trees Regressor, XGB Regressor, LGBM Regressor, K-Neighbors Regressor. Table.5 shows the RMSE value of six ensemble model based on the training and testing data.

Table 5 RMSE value of training and testing data for different ensemble model

Model name	**RMSE value on Training**	**RMSE value on Testing**
Random Forest Regressor	1.57	4.38
Gradient Boosting Regressor	4.67	4.78
Extra Trees Regressor	0.00	4.22
XGB Regressor	2.45	4.52
LGBM Regressor	3.51	4.43
KNeighbors Regressor	5.07	6.07

RMSE value is higher for K-Neighbors regressor both training and testing data set compared with another model. On the other hand, extra trees regressor show 0 RMSE value on the training data meaning that model behaves well on the training dataset. An RMSE of 4.22 here indicates that the model's predictions are, on average, about 4.22 units away from the actual values in the test set. This suggests that while the model performs flawlessly on the training data, it does not perform as well when predicting new, unseen data. The large discrepancy between the training RMSE (0.00) and testing RMSE (4.22) is a classic sign of overfitting. This implies that while the model has learned to perfectly predict or fit the noise and details in the training dataset, it fails to perform effectively on new data.

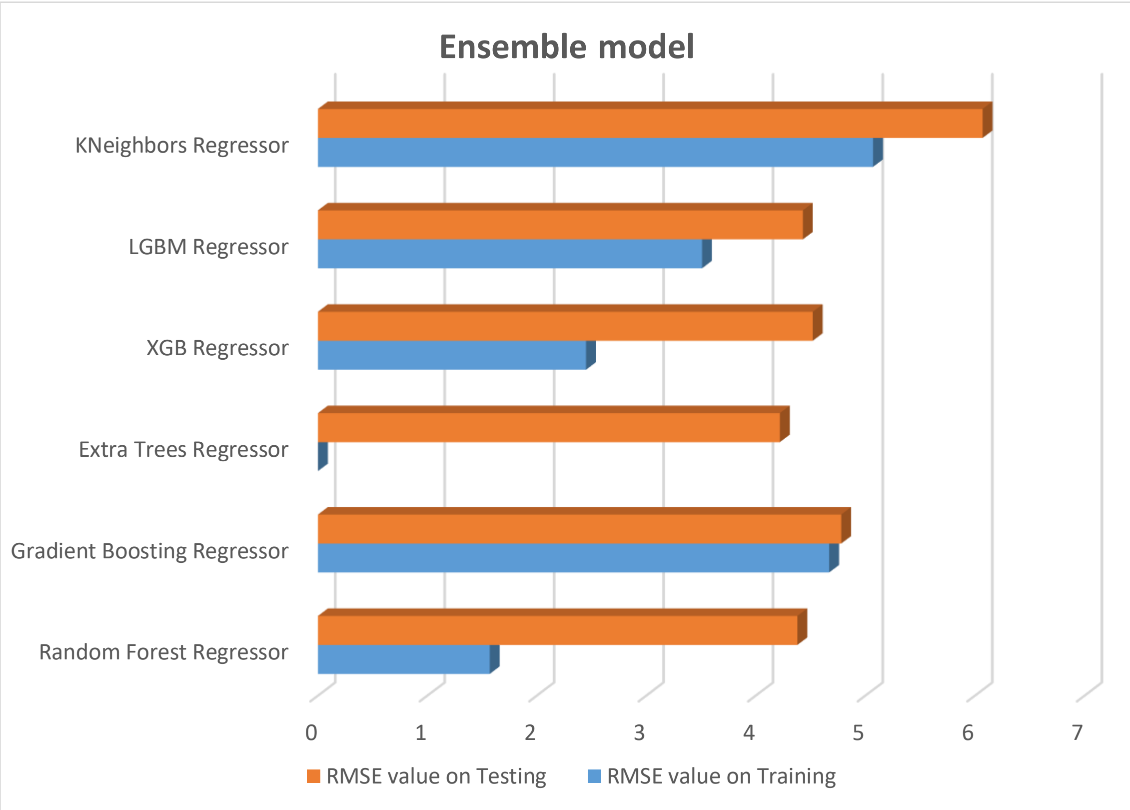

Figure 39. Different ensemble model analysis based on RMSE value

Figure. 40 illustrate the original and the predicted values of electricity consumption obtained by extra trees regressor model. In case of hourly consumption, from this figure that the prediction curves follow the fluctuation of the original electricity consumption.

The predicted values seem to closely follow the trends and fluctuations of the actual data. This indicates that the model is capturing the general pattern of the dataset well. This occurs due to the fact that there are random variations for an individual home user. The prediction appears more accurate in some segments than in others. For example, closer to the end (around 8000 hours), the predictions seem to align more closely with the actual values than in some earlier sections. When the demand changes, the model also reactive in most of the cases but the model couldn't react properly to capture the full intensity or timing of sudden increases.

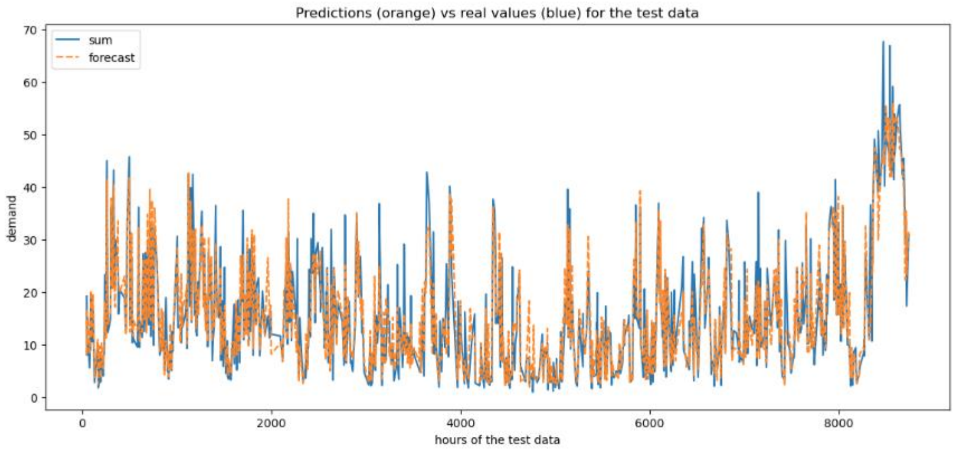

Figure 40. Predicted and real values by extra trees regressor

7.2.3 Neural network

In this section, three different types of neural network are used such as recurrent neural network (RNN), gated recurrent units (GRU) and Least short-term memory (LSTM). Table 6. shows the RMSE value of three different neural network models based on the testing and training data set.

The RMSE value of RNN model is more on training data set compare with the GUR and LSTM model. The results revel, that the RMSE value of training and testing is less in GRU and LSTM compared with RNN. A key aspect in evaluating model performance is how well the model generalizes to new, unseen data (i.e., the testing set).

Table 6 RMSE value of training and testing data for neural network

Model name	RMSE value on Training	RMSE value on Testing
Recurrent neural network (RNN)	3.834	4.323
Gated recurrent units (GRU)	3.688	4.354
Long short-term memory (LSTM)	3.688	4.354

RNN Shows a smaller gap between training and testing RMSE compared to GRU and LSTM, indicating better generalization despite having a slightly higher training RMSE. GRU and LSTM have identical performance metrics with the lowest training RMSE but slightly higher testing RMSE compared to RNN, suggesting they might be slightly overfitting or less effective at generalizing despite being better at fitting the training data.

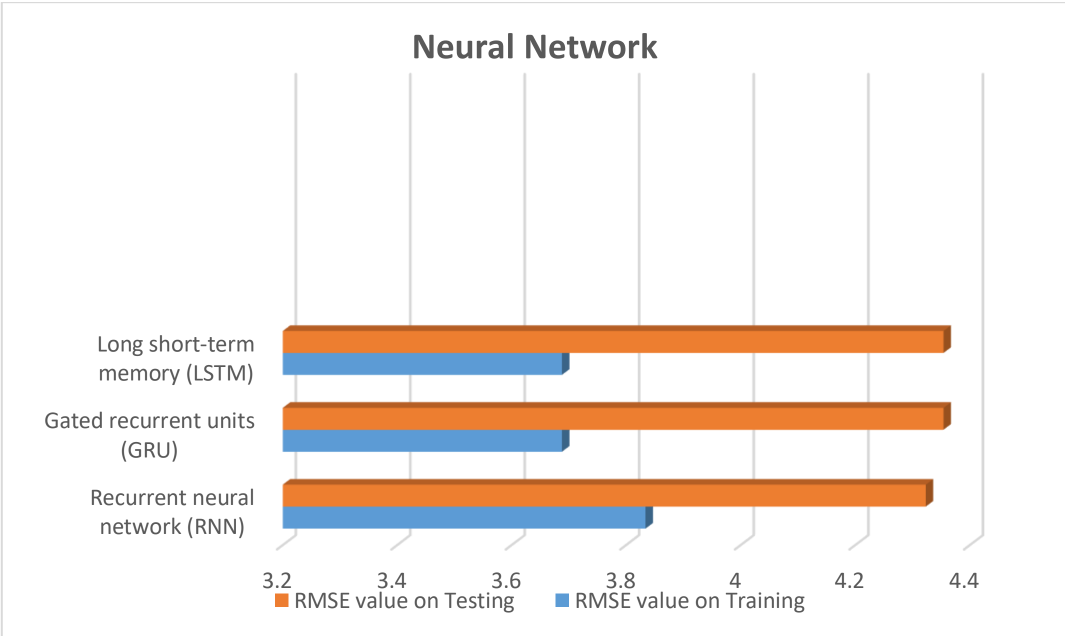

Figure 41. Different neural network analysis based on RMSE value

As shown from these figures 42-44, the three models are able to predict the peaks of the electricity consumption with a very low shift of time compared to the real time. The electricity consumption forecasted by the RNN, GRU and LSTM are very similar to the true data but the LSTM model is better at the prediction of high values.

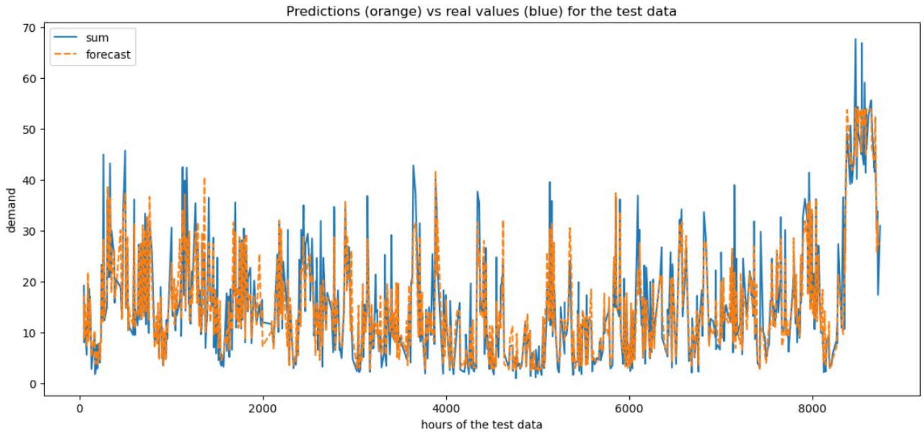

Figure 42. Predicted and real values by recurrent neural network (RNN)

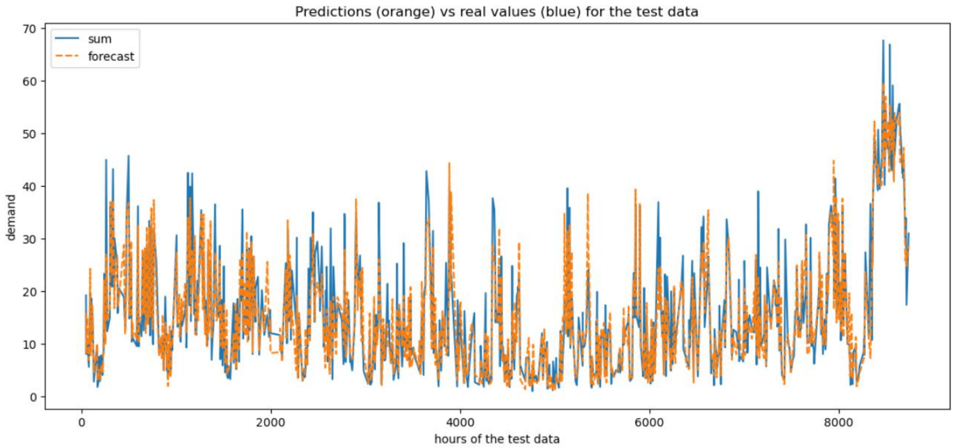

Figure 43. Predicted and real values by gated recurrent units (GRU)

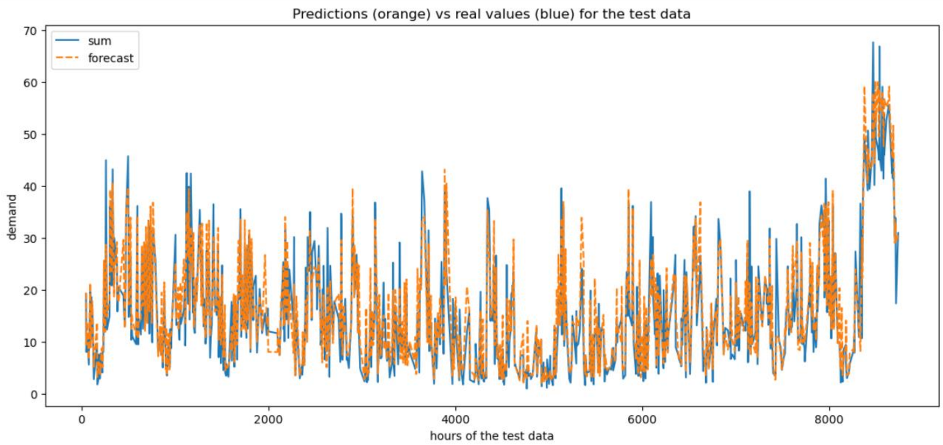

Figure 44. Predicted and real values by long short-term memory (LSTM)

Based on the model configuration, RNN model has 60 epoches, 64 batch size and 48 units. On the other hand, GRU and LSTM both have 50 epoches, 32 batch size and 32 units. However, RNN uses more units and a larger batch size, it indicates that this model a higher capacity model but is still less efficient in capturing the dependencies in data as effectively as GRU or LSTM.

Table 7 Model configuration data for neural network

Model name	Epochs	Batch_size	Units
Recurrent neural network (RNN)	60	64	48
Gated recurrent units (GRU)	50	32	32
Long short-term memory (LSTM)	50	32	32

Moreover, GRU and LSTM are more effective based on the units, batch size and epochs than the RNN. The RNN shows slightly better generalization compared to GRU and LSTM as indicated by the smaller difference between training and testing RMSE.

8 Discussion

The primary objective of this thesis is to examine the significance of energy flexibility from the perspective of consumers, specifically identifying which consumers demonstrate greater flexibility in response to price fluctuations and varying time periods. This study primarily employed several machine learning algorithms to determine which housing exhibits greater adaptability. Furthermore, a versatile consumer is of greater significance in the context of an energy management system. Essentially, demand response mostly centres around energy generation. Furthermore, the primary objective was to determine whether machine learning model exhibited greater accuracy by comparing the root mean square error (RMSE) values of the training and testing data. Conversely, a smaller RMSE number signifies that the model has done effectively with the data, whereas a higher value shows that there is a greater degree of inaccuracy in the training and testing data.

8.1 Limitations of the study

The main limitations of a thesis come in the choice of input data, which holds greater significance than the selection of a model. This study utilises data collected from seven distinct households, with data being recorded every hour. Traditionally, customer-specific predictions were regarded as insignificant because of the unpredictable nature of individual power demands. Each household's unique lifestyle patterns result in varying levels of complexity for the forecasting task. According to the Practice Theory, the energy consumption of residents will reflect their lives, regardless of any inconsistencies, through recurrent patterns. Regarding models, using shorter time intervals results in more accurate and exact predicting. One of the primary constraints of this study is its narrow emphasis on only a few specific aspects.

Moreover, features like temperature and prices are important, successful forecasting for specific dwellings requires the calculation of numerous additional parameters.

8.2 Future study on the field

This work specifically examines and compares application-ready machine learning models available in a high-level programme library. As a result, it serves as a valuable resource for researchers and practitioners seeking effective solutions for forecasting. Nevertheless, there is still potential for additional advancement in accurately predicting which house exhibits greater adaptability. Due to the non-linear nature of our consumer data, it is necessary to utilise more sophisticated machine learning models. In terms of future work, it is possible to develop approaches for parameter adjustment in order to improve the accuracy of forecasting for various client segments, particularly households with significant volatility.

In order to improve the performance of the model, the integration of varied numbers of layers can offer varying levels of abstraction, hence improving the model's ability to learn and perform tasks. This study primarily utilises a limited dataset consisting of one-year data from seven households. However, it is important to consider how the model's behaviour may change when applied to larger datasets. This discovery holds significant implications for economic analysis. Essentially, the load profile remains consistent throughout, thus when the load profile changes, the model's behaviour will be affected. This study relies on real-time data, and the processing of this data in real-time requires substantial computational resources, which can be costly and challenging to handle. In order to make a comparison, it is necessary to execute the model using synthetic data and then compare the obtained findings with those obtained from real-time data.

9 Conclusion

The primary goal of this master's thesis was to identify the most flexible consumer among seven consumers and to identify the most effective machine learning (ML) model for forecasting energy flexibility across various time horizons and scales. In this thesis, eleven different ML models have been used based on the electricity consumption of seven different houses per hour from 2020 to 2021. The research objectives mentioned in Section 1.3 were effectively accomplished by conducting a thorough analysis of the data, utilising various machine learning models, and integrating different model quality metrics with diverse correlation coefficients. This approach resulted in valuable insights and practical solutions for identifying flexible consumers.

In the data set, the main features were air temperature and day-ahead price. Subsequently, to enhance the performance of the ML model, different types of features have been created from the data set, such as month, week, hour, holidays and so on. Mainly, three different types of ML algorithms are used, such as regression analysis, ensemble model and neural networks. For better performance of the ML model, different types of features are used. In the case of regression and ensemble models, six features are used; on the other hand, for neural networks, 26 different features are used to run the model. In this study, we mainly used autocorrelation to find out the lags, which are mainly defined as the previous consumption values taken by a time series at previous time points. In this thesis, different correlation coefficients used for finding the relationship between different features, like the Pearson and Spearman coefficients.

In this case, the data was divided into two groups, such as training and testing data. Furthermore, 90% of the data is used as training data, and 10% is used as testing data. Training data is mainly responsible for learning the data, and testing data focuses on how models behave on unseen data. To evaluate the model's performance, mainly the RMSE performance matrix is used.

The main investigation of this study can be summarised by the primary objective and identifying the most adaptable consumer in terms of energy consumption, considering variations in prices and other factors. Demand response is mostly employed for the purpose of predicting energy usage. To demonstrate the energy adaptability of various consumers, three distinct models, namely RNN, GRU, and LSTM, are employed. According to the three-model flexibility curve, the RNN model indicates that Home 4 exhibits greater flexibility in relation to its price. However, the GRU and LSTM models demonstrate that home 2 has greater adaptability than the other seven houses when the price increases from 10 to 20 euros.

This thesis employed several regression models, including linear regression, Lasso, Ridge, elastic net, and poly-linear regression. The outcome indicated that the ploy-linear regression model exhibited strong performance, as evidenced by the low RMSE value. Among ensemble models such as Random Forest Regressor, Gradient Boosting Regressor, Extra Trees Regressor, XGB Regressor, LGBM Regressor, and K-Neighbours Regressor, the Extra Trees Regressor exhibits a low RMSE value for both testing and training data.

Moreover, in neural networks such as RNN, LSTM, and GRU has been used for testing and evaluating the system while RMSE is larger in RNN compared to LSTM and GRU. As for the evaluation metrics, both the long short-term memory (LSTM) and gated recurrent unit (GRU) models have the same root mean square error (RMSE). When comparing hyperparameters, it is evident that the RNN model employs a greater number of parameters in contrast to the LSTM and GRU models. Essentially, LSTM and GRU are better suited for deploying this data set due to their increased flexibility.

The primary difficulties experienced in this study are the absence of specific characteristics of the households. This is significant because the energy consumption of houses is mostly influenced by factors such as the structure of the house, the number of occupants, the heating system, and weather conditions. While analysing the performances of the developed models, it is important to compute these models on well-labelled data and get some understanding about their behaviours for consumption. This enables the correct monitoring of energy consumption without having to set up and calibrate complex measuring devices. Moreover, it would be crucial to track these houses with regards to various characters and receive updates on electricity usage at any given time.

References

Ahmad, T., Zhang, H., & Yan, B. (2020). A review on renewable energy and electricity requirement forecasting models for smart grid and buildings. Sustainable Cities and Society, 55, 102052.

Albadi, M. H., & El-Saadany, E. F. (2008). A summary of demand response in electricity markets. Electric power systems research, 78(11), 1989-1996.

Alanne, K., & Sierla, S. (2022). An overview of machine learning applications for smart buildings. Sustainable Cities and Society, 76, 103445.

Abumohsen, M., Owda, A. Y., & Owda, M. (2023). Electrical load forecasting using LSTM, GRU, and RNN algorithms. Energies, 16(5), 2283.

Brownlee, J. (2021, April 26). Why use ensemble learning? MachineLearningMastery.com. https://machinelearningmastery.com/why-use-ensemble-learning/

Barhak, J. (2018). Visualization and pre-processing of intensive care unit data using python data science tools. Proceedings from MODSIM World 2018.

Bırant, D., Akça, A., Bozkurt, B., & Bağlan, M. (2022). Classification of scatter plot images using deep learning. Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, 24(71), 631-642.

Bampoulas, A., Pallonetto, F., Mangina, E., & Finn, D. P (2021). A machine learning-based methodology for harnessing the energy flexibility potential of residential buildings.

Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. Machine learning, 3-23.

Cunningham, P. (2022, March 26). Ensembles in Machine Learning - towards Data science. Medium. https://towardsdatascience.com/ensembles-in-machine-learning-9128215629d1

Cortez, V., Rabelo, R., Carvalho, A., Floris, A., & Pilloni, V. (2024). On the Impact of Flexibility on Demand-Side Management: Understanding the Need for Consumer-Oriented Demand Response Programs. International Journal of Energy Research, 2024(1), 8831617.

Chen, Y., Xu, P., Gu, J., Schmidt, F., & Li, W. (2018). Measures to improve energy demand flexibility in buildings for demand response (DR): A review. Energy and Buildings, 177, 125–139. https://doi.org/10.1016/j.enbuild.2018.08.003

El Houda, B. N., Lakhdar, L., & Abdallah, M. (2022, October). Time Series Analysis of Household Electric Consumption with XGBoost Model. In 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS) (pp. 1-6). IEEE.

Firoozi, H., Khajeh, H., & Laaksonen, H. (2021, October). Flexibility Forecast at Local Energy Community Level. In 2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe) (pp. 1-5). IEEE.

Flores, J. H. F., Engel, P. M., & Pinto, R. C. (2012, June). Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting. In *The 2012 International joint conference on neural networks (IJCNN)* (pp. 1-8). IEEE.

Huotari, M., Malhi, A., & Främling, K. (2024). Machine Learning Applications for Smart Building Energy Utilization: A Survey. Archives of Computational Methods in Engineering, 1-20.

Hardmeier, M., Berthold, A., & Siegrist, M. (2024). Factors Influencing People's Willingness to Shift Their Electricity Consumption. Journal of Consumer Policy, 47(2), 199-221.

Hong, T., Pinson, P., Wang, Y., Weron, R., Yang, D., & Zareipour, H. (2020). Energy forecasting: A review and outlook. IEEE Open Access Journal of Power and Energy, 7, 376-388.

Islam, M., Chen, G., & Jin, S. (2019). An overview of neural network. American Journal of Neural Networks and Applications, 5(1), 7-11.

IEEE Standard for Floating-Point Arithmetic—Redline. (2019). IEEE Std 754-2019 (Revision of IEEE 754-2008) - Redline, 1–148.

Jakhar, A. (2017, September). A comprehensive review of power system flexibility. In 2017 IEEE international conference on power, control, signals and instrumentation engineering (ICPCSI) (pp. 1747-1752). IEEE.

Javed, U., Ijaz, K., Jawad, M., Ansari, E. A., Shabbir, N., Kütt, L., & Husev, O. (2021). Exploratory data analysis based short-term electrical load forecasting: A comprehensive analysis. Energies, 14(17), 5510.

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, 10(1), 841-851.

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). Applied linear statistical models. McGraw-hill.

Khajeh, H., Laaksonen, H., Gazafroudi, A. S., & Shafie-Khah, M. (2019). Towards flexibility trading at TSO-DSO-customer levels: A review. Energies, 13(1), 165.

Klimasauskas, C. C. (1989). The 1989 neuro-computing bibliography. *(No Title)*.

Karandish, F., & Šimůnek, J. (2016). A comparison of numerical and machine-learning modeling of soil water content with limited input data. Journal of Hydrology, 543, 892-909.

Lee, S., Seon, J., Hwang, B., Kim, S., Sun, Y., & Kim, J. (2024). Recent Trends and Issues of Energy Management Systems Using Machine Learning. Energies, 17(3), 624.

Lucas, A., Jansen, L., Andreadou, N., Kotsakis, E., & Masera, M. (2019). Load flexibility forecast for DR using non-intrusive load monitoring in the residential sector. Energies, 12(14), 2725.

Li, H., Wang, Z., Hong, T., & Piette, M. A. (2021). Energy flexibility of residential buildings: A systematic review of characterization and quantification methods and applications. Advances in Applied Energy, 3, 100054.

Lund, P. D., Lindgren, J., Mikkola, J., & Salpakari, J. (2015). Review of energy system flexibility measures to enable high levels of variable renewable electricity. Renewable and sustainable energy reviews, 45, 785-807.

Livieris, I. E., Pintelas, E., Stavroyiannis, S., & Pintelas, P. (2020). Ensemble deep learning models for forecasting cryptocurrency time-series. Algorithms, 13(5), 121.

Moghimi, S. M., Gulliver, T. A., & Thirumai Chelvan, I. (2024). Energy Management in Modern Buildings Based on Demand Prediction and Machine Learning—A Review. Energies, 17(3), 555.

Meng, H., Feng, S., & Li, C. (2024). An integrated system of energy generation, storages, and appliances consumption based on machine learning techniques and internet of things. Journal of Energy Storage, 87, 111380.

Michalakopoulos, V., Sarmas, E., Papias, I., Skaloumpakas, P., Marinakis, V., & Doukas, H. (2024). A machine learning-based framework for clustering residential electricity load profiles to enhance demand response programs. Applied Energy, 361, 122943.

McKenna, R., Hernando, D. A., ben Brahim, T., Bolwig, S., Cohen, J. J., & Reichl, J. (2021). Analyzing the energy system impacts of price-induced demand-side-flexibility with empirical data. Journal of cleaner production, 279, 123354.

Mansouri, S. A., Ahmarinejad, A., Javadi, M. S., Nezhad, A. E., Shafie-Khah, M., & Catalão, J. P. (2021). Demand response role for enhancing the flexibility of local energy systems. Distributed energy resources in local integrated energy systems, 279-313.

Maulud, D., & Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning. Journal of Applied Science and Technology Trends, 1(2), 140-147.

Murata, N., Yoshizawa, S., & Amari, S. I. (1992). Learning curves, model selection and complexity of neural networks. *Advances in Neural Information Processing Systems*, *5*.

Mahjoub, S., Chrifi-Alaoui, L., Marhic, B., & Delahoche, L. (2022). Predicting Energy Consumption Using LSTM, Multi-Layer GRU and Drop-GRU Neural Networks. Sensors, 22(11), 4062.

Mukaka, M. M. (2012). A guide to appropriate use of correlation coefficient in medical research. Malawi medical journal, 24(3), 69-71.

Niu, J., Tian, Z., Zhu, J., & Yue, L. (2020). Implementation of a price-driven demand response in a distributed energy system with multi-energy flexibility measures. Energy Conversion and Management, 208, 112575.

Nasteski, V. (2017). An overview of the supervised machine learning methods. Horizons. b, 4(51-62), 56.

Panel, N. P. U. A. (2014). Quick Reference Guide. Cambridge Media, Perth, Australia.

Paridari, K., & Nordström, L. (2020). Flexibility prediction, scheduling and control of aggregated TCLs. Electric Power Systems Research, 178, 106004.

Pra, D. M. (2020, November 2). Time-series-forecasting-with-deep-learning-and-attention-mechanism. Towards Data Science.com

Pandey, R., Khatri, S. K., Singh, N. K., & Verma, P. (Eds.). (2022). Artificial intelligence and machine learning for EDGE computing. Academic Press.

Pedamallu, H. (2024) 'RNN vs GRU vs LSTM - Analytics Vidhya - Medium,' Medium, 22 February. https://medium.com/analytics-vidhya/rnn-vs-gru-vs-lstm-863b0b7b1573.

Pfeifer, A., Herc, L., Bjelić, I. B., & Duić, N. (2021). Flexibility index and decreasing the costs in energy systems with high share of renewable energy. *Energy conversion and management*, *240*, 114258.

Pleil, J. D., Stiegel, M. A., Madden, M. C., & Sobus, J. R. (2011). Heat map visualization of complex environmental and biomarker measurements. Chemosphere, 84(5), 716-723.

Paterakis, N. G., Erdinç, O., & Catalão, J. P. (2017). An overview of Demand Response: Key-elements and international experience. Renewable and Sustainable Energy Reviews, 69, 871-891.

Pham, B. T., Jaafari, A., Avand, M., Al-Ansari, N., Dinh Du, T., Yen, H. P. H., ... & Tuyen, T. T. (2020). Performance evaluation of machine learning methods for forest fire modeling and prediction. Symmetry, 12(6), 1022.

Reiss, P. C., & White, M. W. (2005). Household electricity demand, revisited. *The Review of Economic Studies*, *72*(3), 853-883.

Stulp, F., & Sigaud, O. (2015). Many regression algorithms, one unified model: A review. Neural Networks, 69, 60-79.

Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation coefficients: appropriate use and interpretation. Anesthesia & analgesia, 126(5), 1763-1768.

Ssekulima, E. B., Anwar, M. B., Al Hinai, A., & El Moursi, M. S. (2016). Wind speed and solar irradiance forecasting techniques for enhanced renewable energy integration with the grid: a review. IET Renewable Power Generation, 10(7), 885-989.

Seyedzadeh, S., Rahimian, F. P., Rastogi, P., & Glesk, I. (2019). Tuning machine learning models for prediction of building energy loads. Sustainable Cities and Society, 47, 101484.

Sagheer, A., & Kotb, M. (2019). Time series forecasting of petroleum production using deep LSTM recurrent networks. Neurocomputing, 323, 203-213.

Tyagi, K., Rane, C., & Manry, M. (2022). Regression analysis. In *Artificial intelligence and machine learning for EDGE computing* (pp. 53-63). Academic Press.

Vardakas, J. S., Zorba, N., & Verikoukis, C. V. (2014). A survey on demand response programs in smart grids: Pricing methods and optimization algorithms. IEEE Communications Surveys & Tutorials, 17(1), 152-178.

Veloso, B., Gama, J., Malheiro, B., & Vinagre, J. (2021). Hyperparameter self-tuning for data streams. Information Fusion, 76, 75-86.

Ves, A. V., Ghitescu, N., Pop, C., Antal, M., Cioara, T., Anghel, I., & Salomie, I. (2019, September). A stacking multi-learning ensemble model for predicting near real time energy consumption demand of residential buildings. In 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP) (pp. 183-189). IEEE.

Wen, L., Zhou, K., Yang, S., & Lu, X. (2019). Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting. *Energy*, *171*, 1053-1065.

Wang, Y., Liu, M., Bao, Z., & Zhang, S. (2019). Short-term load forecasting with multi-source data using gated recurrent unit neural networks. Energies, 11(5), 1138.

Wang, Z., & Srinivasan, R. S. (2017). A review of artificial intelligence-based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. Renewable and Sustainable Energy Reviews, 75, 796-808.

Wang, C., Du, W., Zhu, Z., & Yue, Z. (2020). The real-time big data processing method based on LSTM or GRU for the smart job shop production process. Journal of Algorithms & Computational Technology, 14, 1748302620962390

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research, 30(1), 79-82.

Zhou, Z., Qiu, C., & Zhang, Y. (2023). A comparative analysis of linear regression, neural networks and random forest regression for predicting air ozone employing soft sensor models. Scientific Reports, 13(1), 22420.

Appendices

Appendix 1. Information for regression model

1. Data set

	date	h1	h2	h3	h4	h5	h6	h7	Air temperature (degC)	Day-ahead Price [EUR/MWh]
0	2021-01-01 02:00:00	7.697	2.936	3.067	6.959	1.986	4.127	3.002	0.4	24.35
1	2021-01-01 03:00:00	5.661	3.933	3.090	5.268	2.010	4.140	1.391	0.6	23.98
2	2021-01-01 04:00:00	7.128	2.463	4.023	4.922	2.009	4.393	3.379	0.7	23.72
3	2021-01-01 05:00:00	3.593	2.009	3.073	4.851	2.050	1.826	1.725	0.2	23.73
4	2021-01-01 06:00:00	1.248	0.148	3.064	2.733	2.098	1.048	3.186	0.3	24.06

2. Model run code

```
def plot_scores():
    # First, we create three lists with models, the RMSE on the train set and on the test data
    df_score = pd.DataFrame({'model_names' : model_names,
                             'rmse_train' : rmse_train,
                             'rmse_test' : rmse_test})
    #Then, we will change the dataframe shape in a way to have
    #model name in one column (as a variable) and the rmse_train and rmse_test in the other columns (as the values of variabl

    df_score = pd.melt(df_score, id_vars=['model_names'], value_vars=['rmse_train', 'rmse_test'])

    #Then, a barplot is used to show the rmse indexes:

    plt.figure(figsize=(14, 12))
    sns.barplot(y="model_names", x="value", hue="variable", data=df_score)
    plt.show()
```

3. RMSE value of different models

```
LinearRegression          - RMSE on Training  = 8.22 / RMSE on Test = 7.98
Lasso     - RMSE on Training  = 8.27 / RMSE on Test = 8.01
Ridge     - RMSE on Training  = 8.22 / RMSE on Test = 7.98
ElasticNet        - RMSE on Training  = 8.26 / RMSE on Test = 8.01
RandomForestRegressor     - RMSE on Training  = 1.57 / RMSE on Test = 4.38
GradientBoostingRegressor         - RMSE on Training  = 4.67 / RMSE on Test = 4.78
ExtraTreesRegressor       - RMSE on Training  = 0.00 / RMSE on Test = 4.22
XGBRegressor      - RMSE on Training  = 2.45 / RMSE on Test = 4.52
LGBMRegressor     - RMSE on Training  = 3.51 / RMSE on Test = 4.43
KNeighborsRegressor       - RMSE on Training  = 5.07 / RMSE on Test = 6.07
```

Appendix 2. Information of neural network

1. Information about features

```
df_train.iloc[:,15:]
```

	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	lag_8	lag_9	lag_10	...	lag_15	lag_16	lag_17	lag_18	lag_19	lag_20	lag_21	lag_22	la
1696	24.176	18.263	16.769	17.354	18.358	22.298	20.219	18.893	18.240	20.319	...	46.085	47.224	37.810	20.432	21.561	18.397	32.479	19.730	1C
5320	16.873	16.369	9.695	10.666	9.204	7.962	6.089	6.577	4.709	6.795	...	8.356	11.336	11.443	12.777	16.786	24.947	15.114	9.184	14
4958	1.752	3.650	2.184	1.606	2.348	4.023	2.224	5.687	3.010	2.274	...	5.544	10.054	13.846	12.470	4.071	6.862	7.539	4.405	3
5232	11.890	14.221	10.855	16.318	11.915	11.332	8.928	6.545	3.945	2.947	...	4.535	3.865	1.741	3.183	2.601	2.450	5.015	6.907	4
1571	4.400	2.378	3.464	2.052	1.498	2.758	2.847	1.795	2.534	6.473	...	19.146	9.929	9.572	6.072	8.132	5.811	3.504	5.116	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2834	22.099	28.807	27.420	20.496	20.010	16.888	13.824	17.774	15.318	12.839	...	6.189	10.625	6.689	7.356	5.715	10.105	13.048	8.321	1C
5166	9.258	10.594	12.220	12.011	14.945	25.251	19.800	23.747	35.902	32.586	...	18.233	15.031	13.590	12.990	16.839	13.594	16.091	12.604	12
7962	25.223	20.162	19.818	22.331	23.576	17.559	16.758	18.076	17.096	19.495	...	32.248	49.145	40.921	32.125	21.644	35.620	38.919	15.816	12
6100	26.397	25.170	31.641	31.720	36.954	19.639	31.829	17.861	34.864	31.120	...	11.140	13.084	9.653	12.520	12.837	12.489	9.048	10.643	9
7284	9.383	11.350	9.178	6.568	8.912	7.443	9.651	11.754	13.607	15.390	...	16.163	15.487	9.137	19.734	13.094	7.755	11.391	12.811	16

7861 rows × 24 columns

2. RNN model code

```
#pip install tensorflow

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense, Embedding, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.optimizers import SGD


def my_RNN():
    my_rnn = Sequential()
    my_rnn.add(SimpleRNN(units=48, return_sequences=True, input_shape=(X_train.shape[1],1)))
    my_rnn.add(SimpleRNN(units=48, return_sequences=True))
    my_rnn.add(SimpleRNN(units=48, return_sequences=False))
    my_rnn.add(Dense(units=1, activation='linear'))
    return my_rnn
```

```
rnn_model = my_RNN()
rnn_model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
rnn_model.fit(x=X_train, y=y_train, validation_data=(X_test, y_test), epochs=60, batch_size=64)
rnn_model.summary()
```

3. GRU model code

```
from tensorflow.keras.layers import GRU

def my_GRU(input_shape):
    my_GRU = Sequential()
    my_GRU.add(GRU(units=32, return_sequences=True, activation='relu', input_shape=input_shape))
    my_GRU.add(GRU(units=32, activation='relu', return_sequences=False))
    my_GRU.add(Dense(units=1, activation='linear'))
    return my_GRU

gru_model = my_GRU((X_train.shape[1],1))
gru_model.compile(optimizer= Adam(learning_rate=0.001), loss='mean_squared_error')
gru_model.fit(x=X_train, y=y_train, validation_data=(X_test, y_test), epochs=50, batch_size=32)
```

4. LSTM model code

```
from tensorflow.keras.layers import LSTM

def my_LSTM(input_shape):
    my_LSTM = Sequential()
    my_LSTM.add(LSTM(units=32, return_sequences=True, activation='relu', input_shape=input_shape))
    my_LSTM.add(LSTM(units=32, activation='relu', return_sequences=False))
    my_LSTM.add(Dense(units=1, activation='linear'))
    return my_LSTM

lstm_model = my_LSTM((X_train.shape[1],1))
lstm_model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
lstm_model.fit(x=X_train, y=y_train, validation_data=(X_test, y_test), epochs=50, batch_size=32)
```

Appendix 3. RMSE value for all model

```
plt.style.use('fivethirtyeight')
plot_scores()
```